A BOOTSTRAP METPOPOLIS-HASTINGS ALGORITHM FOR BAYESIAN ANALYSIS OF BIG DATA

A Dissertation

by

JINSU KIM

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Faming Liang
Committee Members,	Jianhua Huang
	Huiyan Sang
	Byung-Jun Yoon
Head of Department,	Valen Johnson

December 2014

Major Subject: Statistics

ABSTRACT

Markov chain Monte Carlo (MCMC) methods have proven to be a very powerful tool for analyzing data of complex structures. However, their compute-intensive nature, which typically require a large number of iterations and a complete scan of the full dataset for each iteration, precludes their use for big data analysis. In this thesis, we propose the so-called bootstrap Metropolis-Hastings (BMH) algorithm, which provides a general framework for how to tame powerful MCMC methods to be used for big data analysis; that is to replace the full data log-likelihood by a Monte Carlo average of the log-likelihoods that are calculated in parallel from multiple bootstrap samples. The BMH algorithm possesses an embarrassingly parallel structure and avoids repeated scans of the full dataset in iterations, and is thus feasible for big data problems. Compared to the popular divide-and-conquer method, BMH can be generally more efficient as it can asymptotically integrate the whole data information into a single simulation run. The BMH algorithm is very flexible. Like the Metropolis-Hastings algorithm, it can serve as a basic building block for developing advanced MCMC algorithms that are feasible for big data problems. BMH can also be used for model selection and optimization by combining with reversible jump MCMC and simulated annealing, respectively.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

1. INTRODUCTION

Development of computer technology and fast growing internet have been brought us massive volume of data, such as climate data, biological assay data, website transaction logs, and credit card records. However, such massive data cannot be practically analyzed by a common personal computer becaue their sizes are too big to fit in a memory or it is too time consuming to be analyzed by current statistical methods. To arrange with this problem, one may consider to use parallel and distributed architectures, with multicore and cloud computing platforms providing access to many processors simultaneously, but still it is unclear how to apply current statistical methods to the big data with muticore system. Also, an increase in size typically comes with growth in complexity of data structures. Big data have put a great challange on current statistical methodology.

During past few decades, Markov chain Monte Carlo (MCMC) methods have been widely used in statistical data analysis, and they have proven to be a very powerful and typically unique computational tool for analyzing data of complex structure. However, if a size of the data is too big, it is intractable to run MCMC methods on a matter of memory or could be time consuming because it needs a large numeber of iterations and complete scan of the data set for each iteration. This have been a serious problem of Bayesian approach, whose main weapon is MCMC methods, to big data issue even though it is powerful for complex model. Motivated by success of MCMC methods in analysing data of complex structure, we propose in this thesis a bootstrap Metropolis-Hastings (BMH) algorithm that is feasible for big data and workable on parallel and distributed architectures. Basically, process of BMH algorithm follows that of Metpolis-Hastings (MH) algorithm, which is the general form of MCMC methods. BMH algorithm uses a Monte Carlo average of log likelihoods calculated in certain groups of subsets randomly sampled from the full data set whereas Metropolis-Hastings algorithm uses a log likelihood of the full data set. By taking subsets of data, BMH avoids repeated scan of full data set, and its memory usage can be also controlled.

1.1 Metropolis-Hastings(MH) Algorithm

In a Bayesian approach for data analysis, we investigate posterior distribution of parameters for the model. Let $\boldsymbol{\theta}$ be the parameter vector containing set of parameters in the model, and let $\mathcal{D}$ be the dataset. Then posterior density is

$$\pi\left(\boldsymbol{\theta}|\mathcal{D}\right) = \frac{\pi(\boldsymbol{\theta})f(\mathcal{D}|\boldsymbol{\theta})}{C_1} \tag{1.1}$$

where C_1 is constant, $\pi(\boldsymbol{\theta})$ is prior density of $\boldsymbol{\theta}$, and $f(\mathcal{D}|\boldsymbol{\theta})$ is likelihood of the data given the parameter set $\boldsymbol{\theta}$. It is common that we don't know exact closed form of the posterior density $\pi(\boldsymbol{\theta}|\mathcal{D})$ because it is often hard to find the constant C_1 or because the model(likelihood) is too complicated to integrate it so that it is hard to make inferences for the parameter set. In those cases, one can consider generating samples from the posterior density and making inference from the posterior samples. More we sample from the posterior, the better inference we can get. Metropolis-Hastings algorithm gives Markov chain generated from a certain distributoin function, and its steps are following.

1. *Set initials for* $\boldsymbol{\theta}_t$

2. *Generate candidate* $\boldsymbol{\vartheta}$ *from a proposal density* $g(\cdot|\boldsymbol{\theta}_t)$

3. *Accept* $\boldsymbol{\vartheta}$ *with probability of* α *or reject with remaining probability where* α *is defined by*

$$\alpha = \frac{\pi(\boldsymbol{\vartheta}) f(\mathcal{D}|\boldsymbol{\vartheta}) g(\boldsymbol{\vartheta}|\boldsymbol{\theta}_t)}{\pi(\boldsymbol{\theta}_t) f(\mathcal{D}|\boldsymbol{\theta}_t) g(\boldsymbol{\theta}_t|\boldsymbol{\vartheta})} \tag{1.2}$$

4. *If the candidate is accepted, set the next value as* $\boldsymbol{\theta}_{t+1} = \boldsymbol{\vartheta}$*, or if it is rejected, set* $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t$*.*

5. *Repeat Step 2-4 for* $t = 1, 2, \cdots, B$*, so we have* B *posterior samples* $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_B$*.*

Even though the posterior samples are not independent since they are forming Markov chain, we can have *i.i.d.* samples by taking every m samples where m is enough large to have independence betwwen $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_{t+m}$.

However, still there is an important condition that should be satisfied, that is, $\pi(\boldsymbol{\theta}|\mathcal{D})$ should be possibly evaluated. For the case of big data, it is often too time consuming or infeasible because MCMC take large amount of iterations to guarantee precise inferences from the samples. And also if evaluation of the model is containing an inverse of large matrix, for example multivariate gaussian density, its computational complexity is $O(n^3)$ where n is a number of observations. Hence, increase of a volum of the data will cause lack of memory or seriously long running time.

1.2 Advanced Markov Chain Monte Carlo Methods for Big Data

In the literature, there have been a few methods proposed for big data analysis such as the aggregated estimating equation method (Lin and Xi, 2011), the resampling-based stochastic approximation method (Liang et al., 2013), the bag of little Bootstraps (Kleiner et al., 2014), and the approximate Metropolis-Hastings test(AMHT) method (Korattikra et al., 2014). The aggregated estimating equation method employs divide-and-conquer strategy. It is first to compress the raw data of each partition of the full dataset into some low dimensional statistics, and then to obtain an approximation to the estimating equation estimator, the aggregated estimating equation estimator, by solving and equation aggregated from the saved low dimensional statistics in all partitions. Liang et al. (2013) proposed a new parameter estimator, *maximum mean log-likelihood estimator*, for big data problem, and a resampling-based stochastic approximation method for obtaining such an estimator. The resampling-based stochastic approximation method successfully avoids some difficulties involved in big data problems such as inversion of high dimensional matrix. The bag of little Bootstraps provides an efficient way of bootstrapping for big data estimators which functions by combining the results of bootstrapping multiple small subsets of the big original dataset. We propose in this paper a bootstrap Metropolis-Hastings algorithm that takes advantages of the bag of little Bootstrap and the resampling-based stochastic approximation method. BMH algorithm functions by maximizing adjusted posterior that is a proportional to multiplication of *mean log-likelihood* and prior and uses multiple small bootstrap subsets randomly sampled from the original dataset. In this paper, to show an efficiency of BMH, AMHT and divide-and combind method are implemented, and their results are compared with that of BMH.

In Chapter 2, we briefly describe some recent approaches to solve big data problem, and in Chapter 3, we will see steps of BMH algorithm and its implementation on parallel architecture with some theoretical background. In Chapter 4, we will assess and compare performances of BMH, AMHT, and D&C method using simulated datasets. In Chapter 5, we will apply thses three methods to real datasets, US monthly total precipication, which is spatial data. Finally, in Chapter 6, we close this paper with breif discussion.

2. SOME STRATEGIES FOR BIG DATA

Laney (2012) defined Big data as follows: ”Big data is high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization.” High volume might be explained by large number of observations, high velocity represents fast changing model as time goes, and high variety means that various types of data that needs different approaches to analyze. High velocity and variety cause setting of complex model, and as we briefly said in Chapter 1, Markov chin Monte Carlo is considerably suitable for estimating complex model. Hence, Bayesian approach using MCMC can be one great solution for these types of big data problems.

However, its repeated scan of data makes it computationally too slow to be applied to a data of large amount of observations, which is the case of high volume. In this chapter, we will discuss recent approaches suggested as solutions of big data problems: divide-and-conquer strategy, the approximate Metropolis-Hastings test, a bag of little bootstrap, and a resampling-based stochastic aaproximation.

2.1 Divide-and-Conquer(D&C) Strategy

Lin and Xi (2011) developed a computation and storage efficient algorithm for estimating equation(EE) estimation in massive data set using ”*divide-and-conquer*” strategy. First, one divides full data set into k partitions, and in each partition parameters are estimated using corresponding partitioned data. Then by discarding original data set, one can gaurantee storage efficiency. Later estimated parameters from each partition are gathered to compute aggregated EE, which is weighted average of parameter estimators in each partition. In their paper, Newton Raphson method is used to find maximum likelihood estimator for parameter estimation of each partition, and it is not directly comparable to a bootstrap Metropolis-Hastings(BMH) algorithm, which is bayesian approach. Hence, we brought their strategy, divide-and-conquer, by following steps.

1. *Divide given massive dataset $\mathcal{D}$ into k random partitions, $\mathbf{D}_1, \cdots, \mathbf{D}_k$*

2. *Run MH algorithm using each of partitioned datasets, so we have k chains of $\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_k$*

3. *Calculate $\hat{\boldsymbol{\theta}}_i$ by taking average of the chain in each of partition, i, for $i = 1, \cdots, k$*

4. *Calculate aggregated estimator $\hat{\boldsymbol{\theta}} = \sum_{i=1}^{k} \hat{\boldsymbol{\theta}}_i / k$*

Lin and Xi (2011) used $\mathbf{A}_i = \sum_{j=1}^{n} \frac{\partial \psi(x_{ij}, \hat{\boldsymbol{\theta}}_i)}{\partial \boldsymbol{\theta}}$ as a weight to calculate weighted average of $\hat{\boldsymbol{\theta}}_i$ for $i = 1, \cdots, k$ where $\psi(x_{ij}, \boldsymbol{\theta})$ is score function to be minimized and x_{ij} is j-th observation in i-th partition. However, we instead set $\mathbf{A}_i = 1$ for all $i = 1, \cdots, k$ partitions to have simple computation, and MH algorithm will converges to equalibrium distribution whose maxizer satisfies minimizing the score function when uniform prior is used.

2.2 Approximate Metropolis-Hastings Test(AMHT)

Korattikra et al. (2014) proposed the approximate Metropolis-Hastings test(AMHT) method to generate random samples from the posterior distribution of big data. This method basically develop approximation by reformulating the Metropolis-Hastings(MH) test as a statistical decision problem. First, draw random number $u \sim$ Uniform $[0, 1]$ and in each of MH iteration, with subsample size of n_s, accept the proposal $\boldsymbol{\vartheta}$ if the average difference μ in the log-likelihoods of $\boldsymbol{\vartheta}$ and $\boldsymbol{\theta}_t$ is greater than a threshold μ_0, i.e. compute

$$\mu_0 = \frac{1}{n}\log\left[u\frac{\pi(\boldsymbol{\theta}_t)g(\boldsymbol{\vartheta}|\boldsymbol{\theta}_t)}{\pi(\boldsymbol{\vartheta})g(\boldsymbol{\theta}_t|\boldsymbol{\vartheta})}\right], \quad \text{and} \tag{2.1}$$

$$\mu = \frac{1}{n_s}\sum_{i=1}^{n_s} l_i, \quad \text{where } \ l_i = \log f(x_i|\boldsymbol{\vartheta}) - \log f(x_i|\boldsymbol{\theta}_t) \tag{2.2}$$

Then if $\mu > \mu_0$, accept the proposal and set $\boldsymbol{\theta}_{t+1} = \boldsymbol{\vartheta}$. If $\mu \leq \mu_0$, reject the proposal and set $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t$. This reformation of the MH test makes it easy to frame it as a statistical hypothesis test, $H_0 : \mu > \mu_0$ vs. $H_1 : \mu < \mu_0$. Given μ_0 and a random sample $\left\{l_{i_1}, \cdots, l_{i_{n_s}}\right\}$ drawn without replacement from the population $\{l_1, \cdots, l_n\}$, if the difference between μ_0 and the sample mean $\bar{l} = \sum_{j\in\{i_1,\cdots,i_m\}} l_j/m$ is significantly greater than the standard deviation of $\bar{l}$, we can make the decision to accept or reject the proposal confidently. Otherwise, we should draw more data to increase the precision of $\bar{l}$, i.e. to reduce the standard deviation of $\bar{l}$, until we have enough evidence to make a decision. In summary, with m increasement of subsample, single iteration of AMHT can be achieved by following steps.

1. *Initialize estimated means* $\bar{l} \leftarrow 0$ *and* $\bar{l^2} \leftarrow 0$

2. *Initialize* $n_s \leftarrow 0$*, set* $X_n = \mathcal{D}$

3. *Draw* $u \sim$ *Uniform* $[0, 1]$

4. *Draw mini-batch* $\mathcal{X}$ *of size* $\min(m, n - n_s)$ *without replacement from* X_n *and set* $X_n \leftarrow X_n \setminus \mathcal{X}$

5. *Update* $\bar{l}$ *and* $\bar{l^2}$ *using* $\mathcal{X}$*, and* $n_s \leftarrow n_s + \|\mathcal{X}\|$

6. *Estimate standard deviation* s*, where*

$$s = \frac{s_l}{\sqrt{n_s}}\sqrt{1 - \frac{n_s - 1}{n - 1}} \quad \textit{and} \quad s_l = \sqrt{\left(\bar{l^2} - (\bar{l})^2\right)\frac{n_s}{n_s - 1}} \tag{2.3}$$

7. *Compute* $\delta \leftarrow 1 - \phi_{n_s - 1}\left(\frac{\bar{l} - \mu_0}{s}\right)$ *where* ϕ_k *is CDF of the standard Student-t distribution with* k *degree of freedom*

8. *If* $\delta > \epsilon$*, goto step 4 and repeat, otherwise goto the next step to have decision making*

9. *Accept the proposal so that* $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}'$ *if* $\bar{l} > \mu_0$*, otherwise* $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t$*.*

The advantage of this method is that one can make confident decisions with $n_s \leq n$ data points and save computation time. The bias-variance trade-off can be controlled by adjusting the knob ϵ. When ϵ is high, one makes decisions without sufficient evidence and introduce high bias. As $\epsilon \to 0$, one makes more accurate decisions but is forced to examine more data which results in high variance.

2.3 A Bag of Little Bootstrap(BLB) Method

Kleiner et al. (2014) introduced the Bag of Little Bootstrap(BLB), a new procedure, which incorporates features of both the bootstrap and subsampling to yield a robust, computationally efficient means of assessing the quality of estimators. BLB is well suited to modern parallel and distributed computing architectures and furthermore retains the generic applicability and statistical efficiency of the bootstrap. The BLB fuctions by averaging the results of bootstrapping multiple small subsets of $X_1, X_2, \cdots, X_n$, which are observed *i.i.d.* samples drawn from some (unknown) underlying distribution. More formally, given a subset size $b < n$, BLB samples k subsets of size b from the original n data points, uniformly at random (one can also impose the constrain that the subsets be disjoint). Let $\mathcal{I}_1, \cdots, \mathcal{I}_k \subset \{1, \cdots, n\}$ be the corresponding index multisets (note that $\|\mathcal{I}_j\| = b, \forall j$), and let $\mathbb{P}_{nb}^{(j)} = b^{-1} \sum_{i \in \mathcal{I}_j} \delta_{X_i}$ be the empirical distribution corresponding to subset j where δ_{X_i} is an indicator function that has 1 if $i \in \mathcal{I}_j$ and 0 otherwise. BLB's estimate of $\xi(Q_n(P))$ is then given by

$$k^{-1}\sum_{j=1}^{k}\xi(Q_n(\mathbb{P}_{nb}^{(j)})) \tag{2.4}$$

which is simple average of estimators calculated by observations in each subset.

2.4 A Resampling-based Stochastic Approximation(RSA) Method

Liang et al. (2013) suggested a Resampling-based Stochastic Approximation(RSA) algorithm. In this method, at each iteration, a small subsample is drawn from the full dataset, and then the current estimate of the parameter is updated accordingly under the framework of stochastic approximation. This method also leads to a general parameter estimation approach, maximum mean log-likelihood estimation(MMLE). The method works by minimizing the Kullback-Leibler divergence,

$$\mathrm{KL}(f,g) = -\int \log\left(\frac{f(z|\boldsymbol{\theta})}{g(z)}\right) g(z)dz \tag{2.5}$$

where $f(z|\boldsymbol{\theta})$ is a likelihood function that user specified, and $g(z)$ is unknown true density function. Using subsamples randomly drawn from the given data, $\mathcal{D}$, the Kullback-Leibler divergence can be approximated by

$$\widehat{\mathrm{KL}}(f,g|\mathcal{D}) = C - \binom{n}{m}^{-1}\sum_{i=1}^{\binom{n}{m}} \log f(z_i|\boldsymbol{\theta}) \tag{2.6}$$

where C denotes a constant related to the entropy of $g(z)$, and $\binom{n}{m}$ is the binomial coefficient. Then, the stochastic approximation method is used to estimate $\boldsymbol{\theta}$ by solving the systems of equation $\dfrac{\partial\widehat{\mathrm{KL}}(f,g|\mathcal{D})}{\partial\boldsymbol{\theta}} = -\binom{n}{m}^{-1}\sum_{i=1}^{\binom{n}{m}} H(\boldsymbol{\theta},\mathbf{z}) = 0$, where $H(\boldsymbol{\theta},\mathbf{z}) = \dfrac{\partial \log f(\mathbf{z}|\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}$ is the first order derivative of $\log f(\mathbf{z}|\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$, and $\mathbf{z}$ denotes a random sample drawn from $\mathcal{D}$. Then, assymptotically minimizing

$\widehat{\mathrm{KL}}(f, g|\mathcal{D})$ is to maximize $E(\log f(z_{i_1}, \cdots, z_{i_m}|\boldsymbol{\theta}))$ where $i_1, \cdots, i_m \in \{1, \cdots, n\}$. RSA can be achieved by following steps.

1. *Draw* $\mathbf{z}$ *from* $\mathcal{D}$ *at random and without replacement.*

2. $\boldsymbol{\theta}^{(t+\frac{1}{2})} = \boldsymbol{\theta}^{(t)} + a_{t+1}H(\boldsymbol{\theta}, \mathbf{z})$

3. *If* $\|\boldsymbol{\theta}^{(t+\frac{1}{2})} - \boldsymbol{\theta}^{(t)}\| \leq b$ *then set* $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t+\frac{1}{2})}$, $\pi_{t+1} = \pi_t$, *otherwise set* $\boldsymbol{\theta}^{(t+1)} = T(\boldsymbol{\theta}^{(t)})$ *and* $\pi_{t+1} = \pi_t + 1$

   where π *is number of truncation,* $T : \Theta \to K_0$, K_0 *is compact subset of* Θ *such that initial* $\boldsymbol{\theta}_0 \in K_0$.

For detail explanation of truncation method, please see Liang et al. (2013).

3. A BOOTSTRAP METROPOLIS-HASTINGS(BMH) ALGORITHM

The BLB and RSA brought key idea of the bootstrap Metropolis-Hastings(BMH) algorithm. BMH uses many small size of bootstrap samples and calculate mean log-likelihood with those bootstrap samples, and bring this mean log-likelihood into the Metropolis-Hastings algorithm instead of full data likelihood. The BMH algorithm can be describe as follows.

3.1 Algorithm

Let D_i denote a bootstrap sample of $\mathcal{D}$, which is resampled from the full data set at random and with/without replacement. Let m denote the size of D_i. If resampling is done without replacement, D_i is called a subsample or $\binom{n}{m}$-bootstrap sample. Otherwise, D_i is called m-out-of-n bootstrap sample or m/n-bootstrap sample. Let $\tilde{f}(D_i|\boldsymbol{\theta})$ denote a likelihood-like fuction of D_i, and define

$$l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\theta}) = \frac{1}{k}\sum_{i=1}^{k} \log \tilde{f}(D_i|\boldsymbol{\theta}), \tag{3.1}$$

where k denotes the number of bootstrap samples drawn from $\mathcal{D}$, and $\boldsymbol{D_s} = \{D_1, \cdots, D_k\}$ is the collection of the bootstrap samples. The definition of $\tilde{f}(D_i|\boldsymbol{\theta})$ depends on the feature of $\mathcal{D}$. If the observation in $\mathcal{D}$ are independently and identically distributed ($i.i.d.$), then, regardless D_i is a $\binom{n}{m}$- or m/n- bootstrap sample, we define

$$\tilde{f}(D_i|\boldsymbol{\theta}) = \tilde{f}(x_1^{(i)}, \cdots, x_m^{(i)}|\boldsymbol{\theta}) = \prod_{j=1}^{m} f(x_{ij}^{*}|\boldsymbol{\theta}) \tag{3.2}$$

where x_{ij}^{*} denotes the j-th element in D_i. Since x_{ij}^{*}'s are no longer mutually independent, we say that $\tilde{f}(D_i|\boldsymbol{\theta})$ is a likelihood like function of D_i. For the case that the observations in $\mathcal{D}$ are dependent, how to define $\tilde{f}(D_i|\boldsymbol{\theta})$ will be discussed in section 3.3.

The BMH algorithm works by iterating between the following steps:

1. *Draw $\boldsymbol{\vartheta}$ from a proposal distribution $Q(\boldsymbol{\theta}_t, \boldsymbol{\vartheta})$.*

2. *Draw k bootstrap samples $D_1, \cdots, D_k$ via $\binom{n}{m}$- or m/n- bootstrapping. Let $\boldsymbol{D_s} = \{D_1, \cdots, D_l\}$*

3. *Calculate the BMH ratio:*

$$r(\boldsymbol{\theta}_t, \boldsymbol{D_s}, \boldsymbol{\vartheta}) = \exp\left\{l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\vartheta}) - l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\theta}_t)\right\} \frac{\pi(\boldsymbol{\vartheta})}{\pi(\boldsymbol{\theta}_t)} \frac{Q(\boldsymbol{\vartheta}, \boldsymbol{\theta}_t)}{Q(\boldsymbol{\theta}_t, \boldsymbol{\vartheta})}$$

4. *Set $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\vartheta}$ with probability $\alpha(\boldsymbol{\theta}_t, \boldsymbol{D_s}, \boldsymbol{\vartheta}) = \min\left(1, r(\boldsymbol{\theta}_t, \boldsymbol{D_s}, \boldsymbol{\vartheta})\right)$, and set $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t$ with the remaining probability.*

Regarding this algorithm, we have the following remarks:

- In BMH, $\{\boldsymbol{\theta}_t\}$ form a Markov chain with the transition kernel given by

$$\begin{aligned} P_{m,n,k}(\boldsymbol{\theta}, d\boldsymbol{\vartheta}) = &\sum_{\boldsymbol{D_s} \in \mathbb{D}} \alpha(\boldsymbol{\theta}, \boldsymbol{D_s}, \boldsymbol{\vartheta}) Q(\boldsymbol{\theta}, d\boldsymbol{\vartheta}) \psi(\boldsymbol{D_s}) \\ &+ \delta_{\boldsymbol{\theta}}(d\boldsymbol{\vartheta}) \left[1 - \sum_{\boldsymbol{D_s}' \in \mathbb{D}} \int_{\Theta} \alpha(\boldsymbol{\theta}, \boldsymbol{D_s'}, \boldsymbol{\vartheta}') Q(\boldsymbol{\theta}, d\boldsymbol{\vartheta}') \psi(\boldsymbol{D_s'}) d\boldsymbol{\vartheta}'\right] \end{aligned} \quad (3.3)$$

  where $\mathbb{D}$ denote the space of $\boldsymbol{D_s}$, $\psi(\boldsymbol{D_s})$ denotes the probability of drawing $\boldsymbol{D_s}$, and $\delta_{\boldsymbol{\theta}}(\cdot)$ is an indicator function. For $\binom{n}{m}$-bootsrapping, $\psi(\boldsymbol{D_s}) = \binom{n}{m}^{-k}$; and for m/n-bootstrapping, $\psi(\boldsymbol{D_s}) = 1/n^{mk}$

- Let

  $$P_{m,n,k,\boldsymbol{D_s}}(\boldsymbol{\theta}, d\boldsymbol{\vartheta}) = \alpha(\boldsymbol{\theta}, \boldsymbol{D_s}, \boldsymbol{\vartheta})Q(\boldsymbol{\theta}, \boldsymbol{\vartheta}) + \delta_{\boldsymbol{\theta}}(d\boldsymbol{\vartheta})\left[1 - \int_{\Theta} \alpha(\boldsymbol{\theta}, \boldsymbol{D_s}, \boldsymbol{\vartheta}')Q(\boldsymbol{\theta}, d\boldsymbol{\vartheta}')\right],$$

  denote the transition kernel corresponding to a particular subset data $\boldsymbol{D_s}$. Then $P_{m,n,k}(\boldsymbol{\theta}, d\boldsymbol{\vartheta})$ can be written as a mixture of $P_{m,n,k,\boldsymbol{D_s}}(\boldsymbol{\theta}, d\boldsymbol{\vartheta})$'s; that is,

  $$P_{m,n,k}(\boldsymbol{\theta}, d\boldsymbol{\vartheta}) = \sum_{\boldsymbol{D_s} \in \mathbb{D}} P_{m,n,k,\boldsymbol{D_s}}(\boldsymbol{\theta}, d\boldsymbol{\vartheta})\psi(\boldsymbol{D_s}). \tag{3.4}$$

- When the observations in $\mathcal{D}$ are *i.i.d.*, both the resampling schemes, $\binom{n}{m}-$ or $m/n-$ bootstrapping, can be used in BMH. As shown in section 3.3, the two resampling schemes lead to the same stationary distribution of the Markov chain.

- If we define

  $$\mathbb{H}(\boldsymbol{\theta}|\boldsymbol{D_s}) = -l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\theta}) - \log \pi(\boldsymbol{\theta})$$

  which is the so-called energy function of the posterior distribution, then the BMH ratio can be written as

  $$r(\boldsymbol{\theta}_t, \boldsymbol{D_s}, \boldsymbol{\vartheta}) = \exp\left\{\mathbb{H}(\boldsymbol{\theta}|\boldsymbol{D_s}) - \mathbb{H}(\boldsymbol{\vartheta}|\boldsymbol{D_s})\right\} \frac{Q(\boldsymbol{\vartheta}, \boldsymbol{\theta})}{Q(\boldsymbol{\theta}, \boldsymbol{\vartheta})}$$

- The BMH algorithm consists of a few parameters, namely, m, k, and the proposal distribution. The proposal distribution can be chosen as in the Metropolis-Hasting algorithm; that is, choosing an appropriate proposal distribution such that the resulting BMH chain is irreducible and aperiodic, and the BMH moves have a reasonable acceptance rate, e.g., between 0.2 and 0.4

as suggested by Gelman et al. (1996) for conventional MH algorithms. A formal statement for the requirement of the proposal distribution will be given in condition (B) of Section 3.3.

Since BMH is proposed for simulations on parallel computers, the parameter k species the number of processors/nodes used in computing the averaged log-likelihood function. Theoretically, a large value of k is preferred. However, an extremely large value of k may slow down the computation due to the increased inter-node communications. In our experience, to achieve a good performance for BMH, k does not need to be very large. Both $k = 25$ and $k = 50$ work well for all examples of this paper. The choice of m can depend on the complexity of the model under consideration, in particular, the dimension of $\boldsymbol{\theta}$. In general, m should increase with the complexity of the model.

3.2 Parallel Implementation

We used master/slave apporach. At the begining of the algorithm, The data are read simultaneously at each parallel thread. Initial values and candiates, $\boldsymbol{\theta}_t$ and $\boldsymbol{\vartheta}$, are generated from the master node, and the master node broadcasts (shares) the parameters to all slave nodes. At each of node, subsampling is done independently and simulatenously, and with the subset samples and parameters broadcasted from the master node, log likelihood-like functions, $l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\theta}_t)$ and $l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\vartheta})$, are evaluated. Then, the evaluated $l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\theta}_t)$ and $l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\vartheta})$ are gathered at master node, and at the master node, priors and acceptance rate are calculated so we can decide whether the candidates, $\boldsymbol{\vartheta}$, will be accepted or not. Still, all parallel nodes are having the initial values and the candidates, so again the acceptance indicator from the master node is broadcasted to the all slaves. If the indicator is 1 that represents the candidates are accepted, the next values are updated to the candidates at all parallel nodes, otherwise, the next values are updated to the current values, and hence, all parallel nodes will be having next values regarded as the initial values for the next iteration. For each of iteration of the BMH algorithm, communications are made up for three times, but actually the communication time is not worrisome because the paralle thereads are communicating only scalar and few parameters which is a vector of short length. Steps for BMH algorithm with parallel implementation is as following. Figure 3.1 shows the following steps on a flowchart.

1. *Read data $\mathcal{D}$ simultaneously at every parallel thread.*

2. *Set initial values $\boldsymbol{\theta}_t$ at the master node.*

3. *Generate the candidates $\boldsymbol{\vartheta}$ at the master node.*

4. *Broadcast $\boldsymbol{\theta}_t$ and $\boldsymbol{\vartheta}$ from the master node to the all slaves.*

5. *Draw random subset D_i from the data, $\mathcal{D}$ at i-th parallel thread for all i, then every parallel threads have different random subsets, $\boldsymbol{D_s} = \{D_1, \cdots, D_k\}$.*

6. *At i-th parallel thread, calculate $l_{m,n,k}(D_i|\boldsymbol{\theta}_t)$ and $l_{m,n,k}(D_i|\boldsymbol{\vartheta})$.*

7. *Gather $l_{m,n,k}(D_i|\boldsymbol{\theta}_t)$'s and $l_{m,n,k}(D_i|\boldsymbol{\vartheta})$'s, for all i to the master node.*

8. *At the master node, calculate acceptance ratio, α_{BMH},and decide wheather $\boldsymbol{\vartheta}$ is accepted or not. If accepted, broadcast $\delta = 1$, otherwise braodcast $\delta = 0$ to all the slaves.*

9. *At every parallel threads, update $\boldsymbol{\theta}_{t+1}$ according to the δ broadcasted from the master node so that $\boldsymbol{\theta}_{t+1} = \boldsymbol{\vartheta}$ if $\delta = 1$ or $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t$ if $\delta = 0$.*

10. *Repeat step 3-9 until enough many $\boldsymbol{\theta}$'s are gathered.*

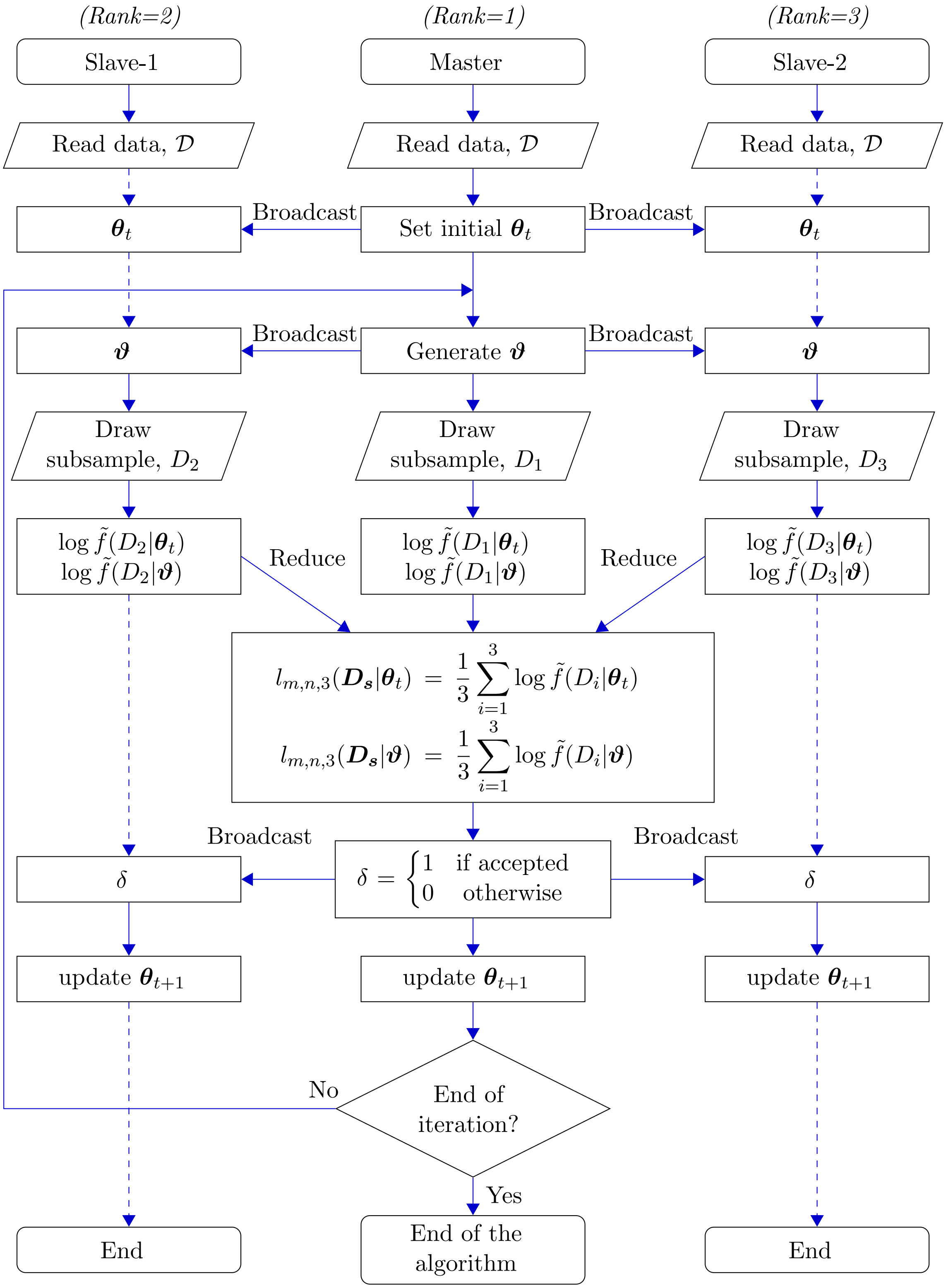
(Rank=2)
(Rank=1)
(Rank=3)
Slave-1
Master
Slave-2
Read data, $\mathcal{D}$
Read data, $\mathcal{D}$
Read data, $\mathcal{D}$
$\boldsymbol{\theta}_t$
Broadcast
Set initial $\boldsymbol{\theta}_t$
Broadcast
$\boldsymbol{\theta}_t$
$\boldsymbol{\vartheta}$
Broadcast
Generate $\boldsymbol{\vartheta}$
Broadcast
$\boldsymbol{\vartheta}$
Draw subsample, D_2
Draw subsample, D_1
Draw subsample, D_3
$\log \tilde{f}(D_2|\boldsymbol{\theta}_t)$
$\log \tilde{f}(D_2|\boldsymbol{\vartheta})$
Reduce
$\log \tilde{f}(D_1|\boldsymbol{\theta}_t)$
$\log \tilde{f}(D_1|\boldsymbol{\vartheta})$
Reduce
$\log \tilde{f}(D_3|\boldsymbol{\theta}_t)$
$\log \tilde{f}(D_3|\boldsymbol{\vartheta})$
$l_{m,n,3}(\boldsymbol{D_s}|\boldsymbol{\theta}_t) = \frac{1}{3}\sum_{i=1}^{3} \log \tilde{f}(D_i|\boldsymbol{\theta}_t)$
$l_{m,n,3}(\boldsymbol{D_s}|\boldsymbol{\vartheta}) = \frac{1}{3}\sum_{i=1}^{3} \log \tilde{f}(D_i|\boldsymbol{\vartheta})$
Broadcast
δ
$\delta = \begin{cases} 1 & \text{if accepted} \\ 0 & \text{otherwise} \end{cases}$
Broadcast
δ
update $\boldsymbol{\theta}_{t+1}$
update $\boldsymbol{\theta}_{t+1}$
update $\boldsymbol{\theta}_{t+1}$
No
End of iteration?
Yes
End
End of the algorithm
End

Figure 3.1: The flowchart of BMH algorithm with 3 processors

3.3 Convergence of the Bootstrap Metropolis-Hastings Algorithm

In this section, we first prove the ergodicity of the BMH algorithm and then discuss how to make Bayesian inference for the full dataset based on the BMH samples. The ergodicity of BMH will be studied in two scenarios, namely, $\binom{n}{m}$-bootstrapping and m/n-bootstrapping.

3.3.1 $\binom{n}{m}$-bootstrapping

To study the ergodicity of BMH, we first assume the following condition holds:

(A) $\sup_{\boldsymbol{\theta}\in\Theta} E|\log f(X|\boldsymbol{\theta})|^2 < \infty$

Conditional on the data set $\mathcal{D}$, $\left\{\log \tilde{f}(D_1|\boldsymbol{\theta}), \cdots, \tilde{f}(D_k|\boldsymbol{\theta})\right\}$ forms a simple random sample without replacement from a finite population. Motivated by this observation, we deine U-statistic

$$U_{m,n}(\mathcal{D}|\boldsymbol{\theta}) = \binom{n}{m}^{-1} \sum_{D_i\in\mathfrak{D}} h(D_i) = \sum_{D_i\in\mathfrak{D}} \log \tilde{f}(D_i|\boldsymbol{\theta}) \tag{3.5}$$

where $\mathfrak{D}$ is the space of D_i and it contains all the possible $\binom{n}{m}$ subsamples of size m, and $h(D_i) = \log \tilde{f}(D_i|\boldsymbol{\theta})$ is called kernel of the U-statistics. Thus, $U_{m,n}(\mathcal{D}|\boldsymbol{\theta})$ is the conditional mean of $\log \tilde{f}(D_1|\boldsymbol{\theta})$ on the dataset $\mathcal{D}$. U-statistics were introduced by Hoeffding (1948), which represent a class of statistics that is especially important in estimation theory. Many well known test statistics and estimators, such as mean and variance, are in fact members of this class. The simple structure of U-statistics has made them widely used for studying general estimation processes such as bootstrapping and jackknifing, and for generalizing those parts of asymptotic theory concerned with sample means. Refer to Lee (1990) for an overview of theory and practice of

U-statistics. By the law of iterated expectations, it is easy to show that

$$\begin{aligned} E(l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\theta}) - U_{m,n}(\mathcal{D}|\boldsymbol{\theta}))^2 &= E\left[E((l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\theta}) - U_{m,n}(\mathcal{D}|\boldsymbol{\theta}))^2|\mathcal{D}\right] \\ &\leq \frac{m}{k} Var(\log f(X|\boldsymbol{\theta})) \end{aligned} \tag{3.6}$$

which, by condition (A) and Chebyshevs inequality, implies that as $k \to \infty$, $n \to \infty$ and $m/k \to 0$,

$$l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\theta}) - U_{m,n}(\mathcal{D}|\boldsymbol{\theta}) \xrightarrow{p} 0, \tag{3.7}$$

where $\xrightarrow{p}$ denotes the convergence in probability. Let $g_m(\mathcal{D}|\boldsymbol{\theta}) = \exp\left\{E\left[\log \tilde{f}(D_i|\boldsymbol{\theta})\right]\right\}$. In the scenario of $\binom{n}{m}$-bootstrapping for $i.i.d$ observations, it follows from (3.2) that

$$g_m(\mathcal{D}|\boldsymbol{\theta}) = \exp\left\{mE\left[\log \tilde{f}(X_1|\boldsymbol{\theta})\right]\right\} \tag{3.8}$$

The variance of a U-statistic based on $i.i.d$ random variables can be expressed in terms of certain conditional expectations. Define for $c = 1, 2, \cdots, m$ the conditional expectation

$$h_c(x_1, \cdots, x_c) = E\left\{\log \tilde{f}(X_1, \cdots, X_m | X_1 = x_1, \cdots, X_c = x_c, \boldsymbol{\theta})\right\},$$

and their variances

$$\sigma_c^2 = Var(h_c(X_1, \cdots, X_c))$$

Then, according to Hoeffding's theorem (see e.g., Lee, 1990, p.12),

$$Var(U_{m,n}) = \binom{n}{m}^{-1} \sum_{c=1}^{m} \binom{m}{c} \binom{n-m}{m-c} \sigma_c^2$$

provided condition (A) holds. Since $\sigma_c^2 = cVar(\log f(X|\boldsymbol{\theta}))$ for the U-statistic defined in (3.5), we have

$$Var(U_{m,n}) = \frac{m^2}{n} Var(\log f(X|\boldsymbol{\theta}))$$

which implies the following theorem holds:

Theorem 3.3.1 *Assume that the condition (A) holds and $m = O(n^\gamma)$, If $\gamma < 1/2$, then*

$$U_{m,n}(\mathcal{D}|\boldsymbol{\theta}) - \log(g_m(\mathcal{D}|\boldsymbol{\theta})) \xrightarrow{p} 0, \quad \text{as} \quad n \to \infty \tag{3.9}$$

Combining (3.7) and (3.9), we have for any $\boldsymbol{\theta} \in \Theta$,

$$l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\theta}) - \log(g_m(\mathcal{D}|\boldsymbol{\theta})) \xrightarrow{p} 0, \quad \text{as} \quad n \to \infty, \tag{3.10}$$

where, as implied by (3.7) and Theorem 3.3.1,

$$m = O(n^\gamma) \quad \text{and} \quad k = O(n^{\gamma+\epsilon_0}) \tag{3.11}$$

for some $\epsilon_0 > 0$ and $\gamma < 1/2$. Define

$$\Gamma_{m,n,k}(\boldsymbol{\theta}, \boldsymbol{D_s}, \boldsymbol{\vartheta}) = \frac{\exp\{l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\vartheta}) - l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\theta})\}}{g_m(\mathcal{D}|\boldsymbol{\vartheta})/g_m(\mathcal{D}|\boldsymbol{\theta})}$$

and

$$\begin{aligned}\lambda_{m,n,k}(\boldsymbol{\theta}, \boldsymbol{D_s}, \boldsymbol{\vartheta}) &= |\log(\Gamma_{m,n,k}(\boldsymbol{\theta}, \boldsymbol{D_s}, \boldsymbol{\vartheta}))| \\ &= |[l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\vartheta}) - \log(g_m(\mathcal{D}|\boldsymbol{\vartheta}))] - [l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\theta}) - \log(g_m(\mathcal{D}|\boldsymbol{\theta}))]|\end{aligned}$$

It follows from (3.10) that

$$\lambda_{m,n,k}(\boldsymbol{\theta}, \boldsymbol{D_s}, \boldsymbol{\vartheta}) \xrightarrow{p} 0, \quad \text{as} \quad n \to \infty \tag{3.12}$$

Define

$$\rho(\boldsymbol{\theta}) = 1 - \sum_{\boldsymbol{D_s} \in \mathbb{D}} \int_{\Theta} \alpha(\boldsymbol{\theta}, \boldsymbol{D_s}, \boldsymbol{\vartheta}) Q(\boldsymbol{\theta}, d\boldsymbol{\vartheta}) \psi(\boldsymbol{D_s}).$$

which represents the mean rejection probability of a BMH move starting from $\boldsymbol{\theta}$.

To establish the convergence of BMH, we also consider the transition kernel

$$P_m(\boldsymbol{\theta}, \boldsymbol{\vartheta}) = \alpha(\boldsymbol{\theta}, \boldsymbol{\vartheta}) Q(\boldsymbol{\theta}, \boldsymbol{\vartheta}) + \delta_{\boldsymbol{\theta}}(d\boldsymbol{\vartheta}) \left[1 - \int_{\Theta} \alpha(\boldsymbol{\theta}, \boldsymbol{\vartheta}') Q(\boldsymbol{\theta}, \boldsymbol{\vartheta}') d\boldsymbol{\vartheta}' \right] \tag{3.13}$$

which is induced by the proposal $Q(\cdot, \cdot)$ for a MH move with the invariant distribution given by

$$\tilde{\pi}_m(\boldsymbol{\theta}|\mathcal{D}) \propto \pi_m(\boldsymbol{\theta}) g_m(\mathcal{D}|\boldsymbol{\theta}) \tag{3.14}$$

Further, we assume the following conditions hold:

(B) Assume that P_m defines an irreducible and aperiodic Markov chain such that $\tilde{\pi}(\cdot) P_m = \tilde{\pi}(\cdot)$. Therefore, for any starting point $\boldsymbol{\theta}_0 \in \Theta$,

$$\lim_{t \to \infty} \| P_m^t(\theta_0, \cdot) - \tilde{\pi}_m(\cdot) \| = 0,$$

where $\|\cdot\|$ denotes the total variation norm.

(C) For any $(\boldsymbol{\theta}, \boldsymbol{\vartheta}) \in \Theta \times \Theta$,

$$0 < \Gamma_{m,n,k}(\boldsymbol{\theta}, \boldsymbol{D_s}, \boldsymbol{\vartheta}) < \infty, \quad \psi(\cdot) - a.s.$$

where $\psi(\boldsymbol{D_s})$ is the resampling probability of $\boldsymbol{D_s}$ from $\mathcal{D}$.

Following from the standard theory of the MH algorithm (see e.g. Tierney, 1994), condition (B) can be simply satisfied by choosing an appropriate proposal distribution $Q(\cdot,\cdot)$. Condition (C) is equivalent to assuming $0 < \exp\{l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\vartheta})$ $-l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\theta})\} < \infty$, which ensures the BMH ratio to be well defined in simulations.

Lemma 3.3.1 states that the kernel $P_{m,n,k}$, defined in (4), has a stationary distribution. Its proof follows the proof of Lemma 1 (except for some notational changes) of Liang and Jin (2013) for the Monte Carlo MH algorithm, where the MH ratio includes a random quantity calculated using Monte Carlo samples. A similar theorem has also been proved in Adrieu and Robert (2009) for the pseudo-marginal approach, where the likelihood function is approximated using a Monte Carlo approach such as the importance sampling method.

Lemma 3.3.1 *Assume conditions (B) and (C) hold. Then for any $m, n, k \in \mathbb{N}$ such that $\rho(\boldsymbol{\theta}) > 0$ for all $\boldsymbol{\theta} \in \Theta$, $P_{m,n,k}$ is also irreducible and aperiodic, and hence there exists a stationary distribution $\hat{\pi}_{m,n,k}(\boldsymbol{\theta}|\mathcal{D})$ such that for any $\boldsymbol{\theta}_0 \in \Theta$,*

$$\lim_{t\to\infty} \|P^t_{m,n,k}(\boldsymbol{\theta}_0,\cdot) - \hat{\pi}_{m,n,k}(\cdot)\| = 0.$$

Lemma 3.3.2 concerns the distance between the kernel $P_{m,n,k}$ and the kernel P_m. It states that the two kernels can be arbitrarily close to each other, provided that k and n are large enough. The proof can be found in the Appendix.

Lemma 3.3.2 *Assume the conditions (A),(B) and (C) hold. If (3.11) holds, then for any $\epsilon \in (0,1]$ and any $\boldsymbol{\theta} \in \Theta$, there exist $N(\boldsymbol{\theta}) \in \mathbb{N}$ and $K(\boldsymbol{\theta}, n) \in \mathbb{N}$ such that* *for any $\phi : \Theta \to [1,1]$ and any $n > N(\boldsymbol{\theta})$ and any $k > K(\boldsymbol{\theta}, n)$,*

$$|P_{m,n,k}\phi(\boldsymbol{\theta}) - P_m\phi(\boldsymbol{\theta})| \leq 4\epsilon.$$

Theorem 3.3.2 concerns the ergodicity of BMH, which states that the kernel $P_{m,n,k}$ shares the same stationary distribution with the kernel P_m when both n and k become large. The proof of this theorem follows from the proof of Theorem 1 of Liang and Jin (2013) with some minor changes for accommodating the double limits for k and n.

Theorem 3.3.2 *Assume the conditions (A), (B) and (C) hold and the observations are i.i.d. If (3.11) holds, then for any $\epsilon \in (0,1]$ and any $\boldsymbol{\theta}_0 \in \Theta$, there exist $N(\epsilon, \boldsymbol{\theta}_0) \in \mathbb{N}$, $K(\epsilon, \boldsymbol{\theta}_0, n) \in \mathbb{N}$, and $T(\epsilon, \boldsymbol{\theta}_0, n, k) \in \mathbb{N}$ such that for any $n > N(\epsilon, \boldsymbol{\theta}_0)$, $k > K(\epsilon, \boldsymbol{\theta}_0, n)$, and $t > T(\epsilon, \boldsymbol{\theta}_0, n, k)$*

$$\|P^t_{m,n,k}(\boldsymbol{\theta}_0, \cdot) - \tilde{\pi}_m(\cdot)\| \leq \epsilon,$$

where $\tilde{\pi}_m(\cdot)$ is the stationary distribution of P_m as defined in (3.14).

Theorem 3.3.2 establishes the convergence of BMH under the setting that the bootstrap samples $\boldsymbol{D_s}$ are updated at each iteration. In practice, to avoid frequent updating of bootstrap samples, one may repeatedly use them for a small number of iterations. This may accelerate BMH, especially when m is large. Let κ_0 denote the number of repeated iterations. Following from (3.4), the transitional kernel of BMH for this repeated bootstrap version can be written as

$$\tilde{P}_{m,n,k}(\boldsymbol{\theta}, d\boldsymbol{\vartheta}) = \sum_{\boldsymbol{D_s} \in \mathbb{D}} P^{\kappa_0}_{m,n,k,\boldsymbol{D_s}}(\boldsymbol{\theta}, d\boldsymbol{\vartheta})\psi(\boldsymbol{D_s}). \tag{3.15}$$

Under the same conditions of Lemma 3.3.2, it is shown in the Appendix that

$$\|\tilde{P}_{m,n,k}\phi(\boldsymbol{\theta}) - P_m^{\kappa_0}\phi(\boldsymbol{\theta})\| \leq 4\kappa_0\epsilon. \tag{3.16}$$

Hence, the convergence established in Theorem 3.3.2 still follows for the BMH algorithm with repeated use of bootstrap samples.

In Section 3.4, we will consider the asymptotics of $\tilde{\pi}_m(\boldsymbol{\theta}|\mathcal{D})$. In particular, we will discuss how $\tilde{\pi}_m(\boldsymbol{\theta}|\mathcal{D})$ is related to the whole data posterior $\pi(\boldsymbol{\theta}|\mathcal{D})$ as m becomes large, and how to make Bayesian inference for $\pi(\boldsymbol{\theta}|\mathcal{D})$ based on the samples simulated by the BMH algorithm.

3.3.2 m/n-bootstrapping

Under this scenario, the ergodicity of the BMH algorithm can be studied in a similar way to $\binom{n}{m}$-bootstrapping. In what follows, we show with appropriate conditions that BMH has the same stationary distribution for the two bootstrapping schemes.

First, we define

$$V_{m,n}(\mathcal{D}|\boldsymbol{\theta}) = n^{-m} \sum_{1\leq j_1,\cdots,j_m\leq n} \log \tilde{f}(X_{j_1},\cdots,X_{j_m}|\boldsymbol{\theta})$$

which is the conditional mean of $l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\theta})$ on the dataset $\mathcal{D}$ and is called a von Mises statistic or V-statistic. A straightforward calculation (see the Appendix for the details) shows that

$$E(l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\theta}) - V_{m,n}(\mathcal{D}|\boldsymbol{\theta}))^2 = \frac{m}{k}\left(1-\frac{1}{n}\right) Var(\log f(X|\boldsymbol{\theta})), \tag{3.17}$$

which, by condition (A) and Chebyshev's inequality, implies

$$l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\theta}) - V_{m,n}(\mathcal{D}|\boldsymbol{\theta}) \xrightarrow{p} 0, \tag{3.18}$$

as $k \to \infty$, $n \to \infty$ and $m/k \to 0$. It follows from (3.2) that $V_{m,n}(\mathcal{D}|\boldsymbol{\theta}) = \frac{m}{n}\sum_{X_i \in \mathcal{D}} \log f(X_i|\boldsymbol{\theta})$. This implies the following theorem:

Theorem 3.3.3 *Assume that the condition (A) holds. Let $m = O(n^\gamma)$. If $\gamma < 1/2$, then*

$$V_{m,n}(\mathcal{D}|\boldsymbol{\theta}) - \log(g_m(\mathcal{D}|\boldsymbol{\theta})) \xrightarrow{p} 0, \quad \textit{as} \quad n \to \infty \tag{3.19}$$

Combining (3.18) and (3.19), we have

$$l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\theta}) - \log(g_m(\mathcal{D}|\boldsymbol{\theta})) \xrightarrow{p} 0, \quad \text{as} \quad n \to \infty,$$

under the setting (3.11). Then, by the same reasoning as Theorem 3.3.2, we have the following theorem:

Theorem 3.3.4 *Assume that the observations are i.i.d., and the conditions (A), (B) and (C) hold. If (3.11) is satisfied, then BMH with m/n-bootstrapping has the same stationary distribution as with $\binom{n}{m}$-bootstrapping.*

3.4 Bayesian Inference

This subsection is organized as follows. In Section 3.4.1, we establish the asymptotic normality of $\tilde{\pi}_m(\boldsymbol{\theta}|\mathcal{D})$. In Section 3.4.2 and Section 3.4.3, we discuss how to estimate the mean and asymptotic covariance matrix of $\pi_n(\boldsymbol{\theta}|\mathcal{D})$, respectively.

3.4.1 Asymptotic Normality of $\tilde{\pi}_m(\boldsymbol{\theta}|\mathcal{D})$

For convenience, we rewrite the full data posterior $\pi(\boldsymbol{\theta}|\mathcal{D})$ by $\pi_n(\boldsymbol{\theta}|\mathcal{D})$; i.e.,

$$\pi_n(\boldsymbol{\theta}|\mathcal{D}) \propto \pi_n(\boldsymbol{\theta}) f(\mathcal{D}|\boldsymbol{\theta}),$$

where $f(\mathcal{D}|\boldsymbol{\theta}) = \prod_{i=1}^n f(x_i|\boldsymbol{\theta})$ denotes the likelihood function of $\mathcal{D}$, and $\pi_n(\boldsymbol{\theta})$ denotes the prior of $\boldsymbol{\theta}$ which may depend on the value of n.

The asymptotic normality of posterior distributions has long been studied in the literature. Walker (1969) presented a straightforward approach to the problem for *i.i.d.* observations. Later, this result was generalized to different statistical models and the conditions were also weakened, see e.g., Dawid (1970), Heyde and Johnstone (1979), and Chen (1985). Among those work, the conditions given in Chen (1985) are very general and flexible. For convenience, we shall present Chen's result as

$$\sqrt{n}(\boldsymbol{\theta}^{(n)} - \boldsymbol{\mu}_n) \xrightarrow{d} N(0, \Sigma_0), \tag{3.20}$$

where $\boldsymbol{\theta}^{(n)}$ denotes a generic sample of the full data posterior $\pi_n(\boldsymbol{\theta}|\mathcal{D})$, $\boldsymbol{\mu}_n$ denotes a local mode of $\pi_n(\boldsymbol{\theta}|\mathcal{D})$,$\xrightarrow{d}$ denotes convergence in distribution, and

$$\begin{aligned}\Sigma_0 &= n \left\{-\partial^2 \log \pi_n(\boldsymbol{\theta}|\mathcal{D})/\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T\right\}^{-1}\Big|_{\boldsymbol{\theta}=\boldsymbol{\mu}_n} \\ &= \left\{-\partial^2 \left[\frac{1}{n}\log \pi_n(\boldsymbol{\theta}) + \frac{1}{n}\sum_{i=1}^n \log f(x_i|\boldsymbol{\theta})\right] /\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T\right\}^{-1}\Bigg|_{\boldsymbol{\theta}=\boldsymbol{\mu}_n}\end{aligned}$$

Here, we have assumed that $\pi_n(\boldsymbol{\theta}|\mathcal{D})$ satisfies appropriate conditions, to be specific, the conditions (E_1)-(E_5) given in Lemma A.0.1 of the Appendix. These conditions require that for each n, $\boldsymbol{\mu}_n$ is a local maximum of $\pi_n(\boldsymbol{\theta}|\mathcal{D})$ such that $\partial \log \pi_n(\boldsymbol{\theta}|\mathcal{D})/\partial\boldsymbol{\theta}|_{\boldsymbol{\theta}=\boldsymbol{\mu}_0} =$

0 and Σ_0 is positive definite, and that for large n, $\pi_n(\boldsymbol{\theta}|\mathcal{D})$ becomes highly peaked and behaves like a normal kernel inside a small neighbourhood of $\boldsymbol{\mu}_n$, and the probability outside the neighbourhood is negligible. As pointed out by Chen (1985), the posterior $\pi_n(\boldsymbol{\theta}|\mathcal{D})$ can be multimodal, and $\boldsymbol{\mu}_n$ does not need to be the global maximum. However, the concentration condition must be satisfied at $\boldsymbol{\mu}_n$; that is, the probability outside a small neighborhood of $\boldsymbol{\mu}_n$ is negligible.

For BMH, $l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\theta})$ works as the log-likelihood function, which, as shown in Sections 3.3.1 and 3.3.2, converges to $g_m(\mathcal{D}|\boldsymbol{\theta})$ in probability under appropriate conditions. Let $\pi_m(\boldsymbol{\theta})$ denote the prior distribution of $\boldsymbol{\theta}$. Then the posterior density function $\tilde{\pi}_m(\boldsymbol{\theta}|\mathcal{D})$ is given by

$$\tilde{\pi}_m(\boldsymbol{\theta}|\mathcal{D}) \propto \pi_m(\boldsymbol{\theta})g_m(\mathcal{D}|\boldsymbol{\theta}) = \exp\left\{m\left[\frac{1}{m}\log\pi_m(\boldsymbol{\theta}) + E\log f(X|\boldsymbol{\theta})\right]\right\}.$$

Let $\tilde{l}_m(\boldsymbol{\theta}) = m\left[\frac{1}{m}\log\pi_m(\boldsymbol{\theta}) + E\log f(X|\boldsymbol{\theta})\right]$. It is assumed that for each m,

(D_1) $\tilde{l}_m(\boldsymbol{\theta})$ is uniformly continuous on the parameter space Θ, and it has a unique global maximum and a finite number of local maxima.

(D_2) At the global maximum of $\tilde{l}_m(\boldsymbol{\theta})$, denoted by $\boldsymbol{\mu}_m$, the following conditions are satisfied:

(i) $\tilde{l}'_m(\boldsymbol{\mu}_m) = \partial\tilde{l}_m(\boldsymbol{\theta})/\partial\boldsymbol{\theta}\Big|_{\boldsymbol{\theta}=\boldsymbol{\mu}_m} = 0$

(ii) $\partial^2\tilde{l}_m(\boldsymbol{\theta})/\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T$ is continuous on Θ, and $\tilde{l}''_m(\boldsymbol{\mu}_m) = \partial^2\tilde{l}_m(\boldsymbol{\theta})/\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T\Big|_{\boldsymbol{\theta}=\boldsymbol{\mu}_m}$ is negative definite.

The uniform continuity condition can be satisfied by restricting Θ to a large compact set, say, $\Theta = [10^{-100}, 10^{100}]^d$, where d is dimension of $\boldsymbol{\theta}$. As a practical matter, this is equivalent to set $\Theta = \mathbb{R}^d$. The maxima $\boldsymbol{\mu}_m$'s may be different for different values of

m. As $m \to \infty$, $\frac{1}{m} \log \pi_m(\boldsymbol{\theta})$ tends to 0, and thus $\boldsymbol{\mu}_m$ converges to the maximum of $E \log f(X|\boldsymbol{\theta})$. It follows from Jensen's inequality that $E_{\boldsymbol{\theta}^*} \log(f(X|\boldsymbol{\theta})/f(X|\boldsymbol{\theta}^*)) \leq 0$ for any $\boldsymbol{\theta} \in \Theta$, where $\boldsymbol{\theta}^*$ denotes the true value of $\boldsymbol{\theta}$ and $E_{\boldsymbol{\theta}^*}$ denotes expectation with respect to $f(x|\boldsymbol{\theta}^*)$. That is, for any $\boldsymbol{\theta} \in \Theta$,

$$E_{\boldsymbol{\theta}^*} \log f(X|\boldsymbol{\theta}) \leq E_{\boldsymbol{\theta}^*} \log f(X|\boldsymbol{\theta}^*).$$

Moreover, this inequality is strict unless $P(f(X|\boldsymbol{\theta}) = f(X|\boldsymbol{\theta}^*)) = 1$. Hence, as $m \to \infty$, $\boldsymbol{\mu}_m$ will converge to $\boldsymbol{\theta}^*$. The uniqueness condition for the global maximum requires $\boldsymbol{\theta}^*$ to be unique. In the case that the uniqueness condition is violated, e.g., in mixture models, the BMH samples can still be used for model inference after applying a label switching procedure (see e.g., Stephens, 2000). Alternatively, one may impose some constraints on $\boldsymbol{\theta}$ such that $\boldsymbol{\theta}^*$ is unique.

Theorem 3.4.1 shows that under conditions (D_1) and (D_2), $\tilde{\pi}_m(\boldsymbol{\theta}|\mathcal{D})$ will converge to a normal density function. Its proof can be found in the Appendix.

Theorem 3.4.1 *Assume that conditions (D_1) and (D_2) hold for each $m > 0$. Then, as $m \to \infty$, we have*

$$\sqrt{m}(\boldsymbol{\theta}^{(m)} - \boldsymbol{\mu}_m) \xrightarrow{d} N(0, \tilde{\Sigma}_0), \quad \textit{as} \quad m \to \infty, \tag{3.21}$$

where $\boldsymbol{\theta}^{(m)}$ denotes a generic sample of $\tilde{\pi}_m(\boldsymbol{\theta}|\mathcal{D})$, and $\tilde{\Sigma}_0 = m\left[\tilde{l}''_m(\boldsymbol{\mu}_m)\right]^{-1}$.

Let $b(\boldsymbol{\theta})$ be a function of $\boldsymbol{\theta}$. Suppose that $\partial b(\boldsymbol{\theta})/\partial \boldsymbol{\theta}$ exists and is not 0. Then, by Delta method, we have

$$\sqrt{m}(b(\boldsymbol{\theta}^{(m)}) - b(\boldsymbol{\mu}_m)) \xrightarrow{d} N\left(0, \left(\frac{\partial b(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^T \tilde{\Sigma}_0 \left(\frac{\partial b(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)\right), \quad \text{as} \quad m \to \infty. \tag{3.22}$$

Further, by the convergence of the averaged observed information to the Fisher information

$$-\frac{1}{n}\sum_{i=1}^{n}\frac{\partial^2 \log f(x_i|\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T} \to -E\left(\frac{\partial^2 \log f(x|\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T}\right)$$

under regularity conditions, we have

$$\|\tilde{\Sigma}_0 - \Sigma_0\| \xrightarrow{p} 0 \quad \text{as} \quad n \to \infty \tag{3.23}$$

That is, Σ_0 can be estimated based on the BMH samples simulated from $\tilde{\pi}_m(\boldsymbol{\theta}|\mathcal{D})$.

As implied by (3.21) and (3.23), BMH has the capability to incorporate the whole data information into a single simulation run. Hence, it can have quite different performance from the D&C method. For the latter, suppose that the dataset has been divided into K subsets $\mathcal{D}_1, \cdots, \mathcal{D}_K$, and each is of size m, i.e., $n = m \times K$. If m is reasonably large, for each subset the corresponding posterior distribution is approximately normal; that is,

$$\sqrt{m}(\boldsymbol{\theta}_i^{(m)} - \boldsymbol{\mu}_{m,i}) \xrightarrow{d} N(0, \Sigma_{m,i}),$$

where i indexes the i-th subset, and $\boldsymbol{\mu}_{m,i}$ and $\Sigma_{m,i}$ denote, respectively, the mean and covariance matrix of the posterior distribution based on the subset $\mathcal{D}_i$. If m is small, $\boldsymbol{\mu}_{m,i}$'s and $\Sigma_{m,i}$'s can be quite different from others. It is true that $\boldsymbol{\mu}_{m,i}$ and $\Sigma_{m,i}$ will asymptotically lose their dependentce on i when m becomes large, but this comes at a price of increasing computational cost. As shown by simulated example in Section 4.1.1, this dependence can lose supprisingly slowly. For a simple linear regression of three predictors, the dependence can still exist for $m = 10^4$, see Table 4.4 for the details.

3.4.2 Estimation of the Mean of $\pi_n(\boldsymbol{\theta}|\mathcal{D})$

First, we explore the relationship between $b(\boldsymbol{\mu}_n)$ and $b(\boldsymbol{\mu}_m)$. Consider the standard Laplace approximation for posterior means, see e.g., Lindley (1961, 1980), Kass et al. (1990) and Miyata (2004) for the details. Given a prior π, a log-likelihood $\log p(x|\boldsymbol{\theta})$, a positive function ξ, and a real function h, we define h_n and ρ by $h_n(\boldsymbol{\theta}) = -\log p(x|\boldsymbol{\theta})/n - \log \xi/n$ and $\rho = \pi/\xi$. Suppose that we are interested in estimating the posterior mean $E_{\pi_n}[b(\boldsymbol{\theta})]$ for an integrable function $b(\boldsymbol{\theta})$, where $E_{\pi_n}[]$ denotes expectation with respect to the posterior $\pi_n(\boldsymbol{\theta}|\mathcal{D})$. Under regularity conditions, $E_{\pi_n}[b(\boldsymbol{\theta})]$ can be approximated as follows:

$$\begin{aligned} E_{\pi_n}[b(\boldsymbol{\theta})] &= \frac{\int_\Theta b(\boldsymbol{\theta})\rho(\boldsymbol{\theta}) \exp\{-nh_n(\boldsymbol{\theta})\}\, d\boldsymbol{\theta}}{\int_\Theta \rho(\boldsymbol{\theta}) \exp\{-nh_n(\boldsymbol{\theta})\}\, d\boldsymbol{\theta}} \\ &= b(\hat{\boldsymbol{\theta}}) + \frac{1}{n}\sum_{ij} b_i h^{ij} \left\{ \frac{\rho_j(\hat{\boldsymbol{\theta}})}{\rho(\hat{\boldsymbol{\theta}})} - \frac{1}{2}\sum_{rs} h^{rs} h_{rsj} \right\} + \frac{1}{2n}\sum_{ij} h^{ij} b_{ij} + O(n^{-2}) \end{aligned} \tag{3.24}$$

where $\rho_j(\hat{\boldsymbol{\theta}}) = \partial\rho(\hat{\boldsymbol{\theta}})/\partial\theta_j$, $b_i = \partial b(\hat{\boldsymbol{\theta}})/\partial\theta_i$, $b_{ij} = \partial^2 b(\hat{\boldsymbol{\theta}})/\partial\theta_i\partial\theta_j$, $h_{rsj} = \partial^3 h_n(\hat{\boldsymbol{\theta}})/\partial\theta_r\partial\theta_s\partial\theta_j$, h^{ij} is the component of the matrix $[\partial^2 h_n(\boldsymbol{\theta})/\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T]^1$, and θ_j denotes the j-th component of $\boldsymbol{\theta}$. There are two special cases for the choices of ξ and ρ: If $\xi = 1$ and $\rho = \pi$, then $\hat{\boldsymbol{\theta}}$ becomes the MLE; and if $\xi = \pi$, then $\rho = 1$ and $\hat{\boldsymbol{\theta}}$ becomes the posterior mode.

Suppose that $\boldsymbol{\theta}$ is subject to the following prior:

$$\pi_m(\boldsymbol{\theta}) = [\pi_n(\boldsymbol{\theta})]^{m/n}. \tag{3.25}$$

Note that this prior setting facilitates the following theoretical analysis, but is not an essential requirement. Then it follows from (3.24) that the posterior mean $E_{\tilde{\pi}_m}[b(\boldsymbol{\theta})]$, which is defined with respect to the posterior $\tilde{\pi}_m(\boldsymbol{\theta}|\mathcal{D})$, can be approximated by

$$
\begin{aligned}
E_{\tilde{\pi}_m}[b(\boldsymbol{\theta})] &= \frac{\int_\Theta b(\boldsymbol{\theta})\rho(\boldsymbol{\theta})\exp\{-mE_h(\boldsymbol{\theta})\}\,d\boldsymbol{\theta}}{\int_\Theta \rho(\boldsymbol{\theta})\exp\{-mE_h(\boldsymbol{\theta})\}\,d\boldsymbol{\theta}} \\
&= \frac{\int_\Theta b(\boldsymbol{\theta})\rho(\boldsymbol{\theta})\exp\{-mh_n(\boldsymbol{\theta})\}\,d\boldsymbol{\theta}}{\int_\Theta \rho(\boldsymbol{\theta})\exp\{-mh_n(\boldsymbol{\theta})\}\,d\boldsymbol{\theta}} \\
&= b(\hat{\boldsymbol{\theta}}) + \frac{1}{m}\sum_{ij} b_i h^{ij}\left\{\frac{\rho_j(\hat{\boldsymbol{\theta}})}{\rho(\hat{\boldsymbol{\theta}})} - \frac{1}{2}\sum_{rs} h^{rs}h_{rsj}\right\} + \frac{1}{2m}\sum_{ij} h^{ij}b_{ij} + O(m^{-2}),
\end{aligned}
\tag{3.26}
$$

where ρ_j, b_i, b_{ij}, h_{rsj}, and h^{ij} are as defined in (3.24), $E_h(\boldsymbol{\theta}) = -E\log f(X|\boldsymbol{\theta}) - \log(\xi)/n$, and the second equality follows from the convergence

$$
\frac{1}{n}\sum_{i=1}^{n}\log f(x_i|\boldsymbol{\theta}) \overset{a.s.}{\to} E\log f(X|\boldsymbol{\theta}),
$$

which holds under condition (A). Hence, $E_{\tilde{\pi}_m}[b(\boldsymbol{\theta})] \to b(\hat{\boldsymbol{\theta}})$ as $m \to \infty$. As n goes to infinity, m will also go to infinity, then it follows from (3.24) and (3.26) that

$$
E_{\tilde{\pi}_m}[b(\boldsymbol{\theta})] - E_{\pi_m}[b(\boldsymbol{\theta})] \to 0, \quad \text{as} \quad m, n \to \infty \tag{3.27}
$$

and, by setting $\xi = \pi$ and $\rho = 1$ in (3.24) and (3.26),

$$
\|b(\boldsymbol{\mu}_m) - b(\boldsymbol{\mu}_n)\| \to 0, \quad \text{as} \quad m, n \to \infty. \tag{3.28}
$$

Equation (3.27) suggests that we can use the sample average

$$
\widehat{E_{\tilde{\pi}_m}[b(\boldsymbol{\theta})]} = \frac{1}{T}\sum_{t=1}^{T} b(\boldsymbol{\theta}_t), \tag{3.29}
$$

to estimate $E_{\pi_n}[b(\boldsymbol{\theta})]$, where $\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_T$ denote T samples simulated by BMH from the approximate posterior $\tilde{\pi}_m(\boldsymbol{\theta}|\mathcal{D})$. It follows from (3.27) and the property of MCMC that $\widehat{E_{\tilde{\pi}_m}[b(\boldsymbol{\theta})]}$ provides a consistent estimator for $E_{\pi_n}[b(\boldsymbol{\theta})]$. Further, it follows from (3.24) and the consistency of MLE that $\widehat{E_{\tilde{\pi}_m}[b(\boldsymbol{\theta})]}$ is consistent for $b(\boldsymbol{\theta}^*)$; that is,

$$\widehat{E_{\tilde{\pi}_m}[b(\boldsymbol{\theta})]} \xrightarrow{p} b(\boldsymbol{\theta}^*), \quad \text{as} \quad m \to \infty. \tag{3.30}$$

In practice, m cannot be very large for the reason of computational efficiency. As implied by (3.26), we can improve the accuracy of the estimator of $b(\boldsymbol{\theta}^*)$ using an extrapolation method by fitting the linear regression

$$\widehat{E_{\tilde{\pi}_m}[b(\boldsymbol{\theta})]} = \beta_0 + \beta_1/m + \epsilon$$

for a small set of m, where $1/m$ works as the explanatory variable, and ϵ is the normal random error. Then $\hat{\beta}_0$, the least square estimator of β_0, will serve as an estimator of $b(\boldsymbol{\theta}^*)$, which corresponds to the limit $m \to \infty$.

3.4.3 Estimation of the Covariance Matrix of $\pi_n(\boldsymbol{\theta}|\mathcal{D})$

In addition to the mean of the posterior $\pi_n(\boldsymbol{\theta}|\mathcal{D})$, the asymptotic covariance matrix of $\pi_n(\boldsymbol{\theta}|\mathcal{D})$ can also be simply estimated based on the BMH samples. It follows from (3.21) and (3.23) that $m\hat{\Sigma}_m$ provides a consistent estimator of Σ_0 , where Σ_m denotes the covariance matrix of $\boldsymbol{\theta}^{(m)}$ calculated based on the BMH samples. In summary, BMH provides a simple way to asymptotically integrate the whole data information into a single simulation run and thus a convenient way for Bayesian analysis of big data.

4. SIMULATION STUDIES

4.1 A Linear Regression Example

Consider the normal linear regression

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \epsilon_i, \quad i = 1, \cdots, n$$

where $(\beta_0, \beta_1, \beta_2, \beta_3) = (2, 0.25, 0.25, 0)$ are regression coefficients, and $\epsilon_1, \cdots, \epsilon_n$ are *i.i.d.* normal random errors with mean 0 and variance σ^2, In simulations, we set $n = 10^5$ and $\sigma^2 = 0.25$, generate both $\boldsymbol{x}_1 = (x_{11}, \cdots, x_{n1})^T$ and $\boldsymbol{x}_2 = (x_{12}, \cdots, x_{n2})^T$ from the multivariate normal distribution $N(0,$
$boldsymbolI_n)$, and set $\boldsymbol{x}_3 = (x_{13}, \cdots, x_{n3})^T = 0.7\boldsymbol{x}_2 + 0.3\boldsymbol{z}$, where
$boldsymbolI_n$ is an n-by-n identity matrix, and $\boldsymbol{z}$ is also generated from $N(0,$
$boldsymbolI_n)$. Under this setting, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ are highly correlated with a theoretial correlation corefficient of 0.919. The high correlation between $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ makes the posterior distribution $\pi_n(\boldsymbol{\theta}|\mathcal{D})$ multimodal and the estimators of β_2 and β_3 are negatively correlated. Let $\boldsymbol{\theta} = (\beta_0, \beta_1, \beta_2, \beta_3, \sigma^2)$ and $\boldsymbol{\theta}^* = (2, 0.25, 0.25, 0, 0.25)$ be its true value. We will use this example to demonstrate that (i) BMH can be used for Bayesian analysis of big data; that is, it can correctly estimate the mean and covariance matrix of $\pi_n(\boldsymbol{\theta}|\mathcal{D})$, and (ii) the multimodality of $\pi_n(\boldsymbol{\theta}|\mathcal{D})$ does not affect the asymptotically normality of $\tilde{\pi}_m(\boldsymbol{\theta}|\mathcal{D})$. For this example, $\boldsymbol{\theta}^*$ is unique, but the posterior can contain two separated modes.

To conduct Bayesian analysis for this example, we let $\boldsymbol{\theta}$ be subject to the following prior distribution

$$\pi_m(\boldsymbol{\theta}) \propto \left(\frac{1}{\sigma}\right)^{m/n}$$

as suggested in Section 3.4. To explore the performance of BMH with different values of k and m, we tried all cross settings of $k = 25, 50$ and $m = 200, 500, 1000$. For each setting of (k, m), BMH was run for 20 times independently; 10 runs for $\binom{n}{m}$-bootstrapping and 10 runs for m/n-bootstrapping. Each run consisted of 55,000 iterations, where the first 5,000 iterations were discarded for the burn-in process and the samples generated in the remaining iterations were used for parameter estimation. To facilitate simulations, we have reparameterized σ^2 by $\log(\sigma^2)$. Denote the reparameterized parameter vector by $\tilde{\boldsymbol{\theta}}$.

The proposal distribution consisted of two equally weighted components. The first component is designed according to the hit-and-run algorithm (Chen and Schmeiser, 1996), which is to set

$$\tilde{\boldsymbol{\theta}}' = \tilde{\boldsymbol{\theta}}_t + S \times \boldsymbol{e}$$

where $\tilde{\boldsymbol{\theta}}_t$ and $\tilde{\boldsymbol{\theta}}'$ denote, respectively, the current and proposed values of $\tilde{\boldsymbol{\theta}}$, $\boldsymbol{e}$ is a random direction drawn uniformly from a unit sphere, and $S \sim N(0, s^2)$. Here s is called the step size of the proposal. The second component is to randomly choose two components of $\tilde{\boldsymbol{\theta}}_t$ to undergo the modification

$$\tilde{\boldsymbol{\theta}}'_j = \tilde{\boldsymbol{\theta}}_{t,j} + \tilde{\boldsymbol{e}}$$

where $\tilde{\boldsymbol{\theta}}_{t,j}$ denotes the selected components of $\tilde{\boldsymbol{\theta}}_t$ and $\tilde{\boldsymbol{e}} \sim N(0, s^2\boldsymbol{I}_2)$. In all simulations of this subsection, we set $s = 0.15$. The resulting acceptance rate of the BMH moves ranges from 0.10 to 0.26 for different values of m. When m is large, the posterior distribution becomes highly peaked. To maintain a reasonable acceptance rate, s should be decreased accordingly. For simplicity, we fix $s = 0.15$ in the every simulations of this section.

Table 4.1: Prameter estimation results of MH and BMH for the simulated example. The numbers in paranthesis denote the standard deviations of the estimates, which are calculated by average over 10 independent runs. The true value of $(\beta_0, \beta_1, \beta_2, \beta_3, \log\sigma^2)$ is (2.0,0.25,0.25,0,-1.3863)

(k, m)	$\frac{m}{n} \times 100\%$	β_0	β_1	β_2	β_3	$\log\sigma^2$
MH for the full data						
$(1, 10^5)$	100%	1.9997 $(5.2e^{-6})$	0.2502 $(4.9e^{-6})$	0.2491 $(8.3e^{-6})$	0.0001 $(1.6e^{-5})$	-1.3852 $(1.9e^{-5})$
BMH with $\binom{n}{m}$-bootstrapping						
$(25, 200)$	0.2%	2.0028 $(2.6e^{-4})$	0.2531 $(2.7e^{-4})$	0.2576 $(1.1e^{-3})$	-0.0115 $(1.6e^{-3})$	-1.3619 $(1.0e^{-3})$
$(25, 500)$	0.5%	2.0029 $(1.3e^{-4})$	0.2531 $(2.0e^{-4})$	0.2556 $(1.9e^{-3})$	-0.0087 $(2.4e^{-3})$	-1.3771 $(1.0e^{-3})$
$(25, 1000)$	1.0%	2.0030 $(2.2e^{-4})$	0.2532 $(1.8e^{-4})$	0.2571 $(7.8e^{-4})$	-0.0105 $(1.1e^{-3})$	-1.3827 $(7.5e^{-4})$
$(50, 200)$	0.2%	1.9998 $(1.4e^{-4})$	0.2504 $(1.3e^{-4})$	0.2463 $(4.2e^{-4})$	0.0041 $(5.9e^{-4})$	-1.3600 $(3.9e^{-4})$
$(50, 500)$	0.5%	1.9998 $(8.3e^{-5})$	0.2503 $(7.6e^{-5})$	0.2497 $(2.0e^{-4})$	-0.0009 $(3.3e^{-4})$	-1.3753 $(2.4e^{-4})$
$(50, 1000)$	1.0%	1.9998 $(3.9e^{-5})$	0.2500 $(5.1e^{-5})$	0.2489 $(1.2e^{-4})$	-0.0007 $(2.1e^{-4})$	-1.3807 $(2.1e^{-4})$
BMH with m/n-bootstrapping						
$(25, 200)$	0.2%	2.0027 $(3.1e^{-4})$	0.2532 $(2.3e^{-4})$	0.2514 $(2.0e^{-3})$	-0.0026 $(2.5e^{-3})$	-1.3619 $(1.1e^{-3})$
$(25, 500)$	0.5%	2.0028 $(2.1e^{-4})$	0.2533 $(1.4e^{-4})$	0.2565 $(9.4e^{-4})$	-0.0096 $(1.1e^{-3})$	-1.3788 $(4.4e^{-4})$
$(25, 1000)$	1.0%	2.0030 $(1.3e^{-4})$	0.2533 $(1.1e^{-4})$	0.2569 $(6.5e^{-4})$	-0.0106 $(9.2e^{-4})$	-1.3832 $(6.6e^{-4})$
$(50, 200)$	0.2%	2.0030 $(2.4e^{-4})$	0.2529 $(2.8e^{-4})$	0.2560 $(2.1e^{-3})$	-0.0091 $(2.8e^{-3})$	-1.3616 $(6.1e^{-4})$
$(50, 500)$	0.5%	2.0030 $(1.9e^{-4})$	0.2530 $(1.4e^{-4})$	0.2552 $(1.1e^{-3})$	-0.0078 $(1.6e^{-3})$	-1.3776 $(6.3e^{-4})$
$(50, 1000)$	1.0%	2.0029 $(1.2e^{-4})$	0.2531 $(1.5e^{-4})$	0.2569 $(1.2e^{-3})$	-0.0108 $(1.6e^{-3})$	-1.3813 $(8.0e^{-4})$

For comparison, Bayesian analysis has also been done for the model using the full dataset with the prior $\pi_n(\boldsymbol{\theta}) \propto 1/\sigma$. The MH algorithm was run for 10 times independently with the full dataset. Each run consisted of 55,000 iterations, where the first 5,000 iterations were discarded for the burn-in process and the samples generated in the remaining iterations were used for inference. The proposal distribution used in these runs is the same as that used in the BMH runs. Table4.1 compare the parameter estimates produced by MH and BMH. The comparison confirms the validity of BMH: The two resampling schemes, $\binom{n}{m}$-bootstrapping and m/n-bootstrapping, result in almost the same estimates, and the estimates tend to converge to the MH estimates as k and m become large. Note that the standard deviations of the BMH estimates tend to decrease as k and m increase. Table4.1 also shows that BMH is quite robust to the choice of k and m; it can work well with k as low as 25 and a wide range of m.

As discussed in Section 3.4.2, the BMH estimates can potentially be improved via extrapolation. To illustrate this procedure, we fit a linear regression for the BMH estiamtes of $\log(\sigma^2)$, obtained with (k, m)=(50,200), (50,500), and (50,1000), versus $1/m$. Figure 4.1 shows the scatter plot of the BMH estimates and the fitted regression line

$$\widehat{\log(\sigma^2)} = -1.38577 + 5.1541/m$$

whose coefficient of determination is R^2=0.7031. The extrapolated estiamte of $\log(\sigma^2)$ at $m = n$ is -1.385721, which is surprisingly close to the MH estimate -1.3852.

Next, we explore the estimation of Σ_0 , the asymptotic covariance matrix of $\pi_n(\boldsymbol{\theta}|\mathcal{D})$, using BMH. To estimate Σ_0 , we thinned the MH and BMH runs by a factor

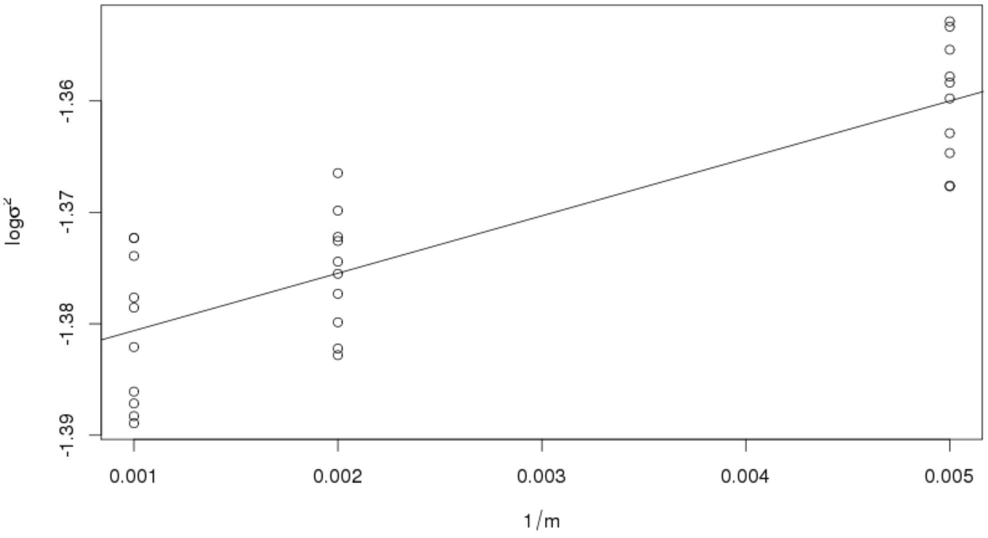

Figure 4.1: Regression extrapolation for the BMH estimates of $\log(\sigma^2)$ obtained with (k, m)=(50,200), (50,500), and (50,1000): The fitted line is $\widehat{\log(\sigma^2)} = -1.38577 + 5.1541/m$

of 500 such that the resulting samples are approximately mutually independent. Note that the samples obtained in BMH runs are usually less correlated than those obtained in MH runs, as $\tilde{\pi}_m(\boldsymbol{\theta}|\mathcal{D})$ is less highly peaked than $\pi_n(\boldsymbol{\theta}|\mathcal{D})$. But, for simplicity, we thinned both by the same factor. Table 4.2 summarizes the estimates of Σ_0 obtained by MH and BMH (with m/n-bootstrapping and $k = 50$). The BMH estimates obtained under other settings are similar. In this table, we report the mean and standard deviations of the estimates of $\sigma_{11}^2, \cdots, \sigma_{55}^2$, σ_{12}^2 and σ_{34}^2 obtained in 10 independent runs by the respective algorithms, where σ_{ij}^2 denotes the (i, j)th element of Σ_0 . The elements $\sigma_{ii}, i = 1, \cdots, 5$, correspond to the posterior variances of $\beta_0, \cdots, \beta_3$ and $\log \sigma^2$, respectively. The element σ_{34}^2 corresponds to the posterior covariance of β_2 and β_3, which are known to be negatively correlated. The element

Table 4.2: Mean and standard deviations (in the paranthesis) of the estimates of $\sigma^2_{11}, \cdots, \sigma^2_{55}, \sigma^2_{12}$ and σ^2_{34} obstained by MH and BMH (with k=50 and m/n-bootstrapping) in 10 independent runs, where σ^2_{ij} denotes the (i,j)th elements of Σ_0.

m	σ^2_{11}	σ^2_{22}	σ^2_{33}	σ^2_{44}	σ^2_{55}	σ^2_{12}	σ^2_{34}
MH for the full data							
	0.259	0.243	1.521	2.652	2.133	0.011	-1.836
	(0.039)	(0.027)	(0.171)	(0.383)	(0.215)	(0.024)	(0.245)
BMH with m/n-bootstrapping							
200	0.274	0.270	1.670	2.840	2.057	-0.014	-1.984
	(0.040)	(0.017)	(0.225)	(0.378)	(0.389)	(0.026)	(0.279)
500	0.268	0.253	1.770	2.978	2.149	0.001	-2.104
	(0.039)	(0.027)	(0.334)	(0.405)	(0.302)	(0.026)	(0.361)
1000	0.289	0.259	1.665	2.815	2.178	0.006	-1.987
	(0.023)	(0.045)	(0.169)	(0.384)	(0.371)	(0.019)	(0.253)

σ^2_{12} corresponds to the posterior covariance of β_0 and β_1 , which are known to be uncorrelated. Note that the MH estimate of Σ_0 is $n\hat{\Sigma}_n$, and the BMH estimate of Σ_0 is $m\hat{\Sigma}_m$, where $\hat{\Sigma}_n$ and $\hat{\Sigma}_m$ are calculated using the thinned MCMC samples from their respective runs. Table 4.2 confirms the validity of BMH for Bayesian inference of big data: It can be used through rescaling to quantify the uncertainty of the estimators corresponding to the full data.

4.1.1 A Comparison Study with Existing Methods

In this section, we compare BMH with two existing methods, the divide-and-conquer (D&C) strategy and approximate MH test (AMHT) methods.

As a natural methodology, the D&C method has often been used in big data analysis. The D&C method used in this thesis proceeds as follows.

We first divide the whole dataset into 50 subsets, each consisting of 2,000 observations. The MH algorithm was then run for each subset data for a total of 55,000 iterations, where the first 5,000 iterations were discarded for the burn-in process, and the remaining 50,000 samples were collected for statistical inference of the model. The proposal used in the simulations was the same as that used by BMH in Section 4.1. The parameters were estimated based on the samples collected from the simulations for each subset data. Finally, we combine all the estimates from each subset data by simply averaging to get the final estimate of the parameters. D&C method is also applied to the same cluster architecture that BMH was appled to. After we divide the data set, all process run independently until the final iteration of the chain. So, there are minimum number of comunications between parallel threads. Natually, D&C method takes more computational time with fixed number of parallel threads because it scans 10 times larger number of subjects at every iteration in each group whereas we can fix smaller number of samples, $m = 200$, for BMH algorithm.

Korattikra et al. (2014) proposed an approximate MH test (AMHT) method for sampling from the posterior distribution of big data. As described in Chapter 2, the significance level ϵ controls the approximate accuracy of the posterior distribution and also the proportion of the data to be used at each iteration of the algorithm. To compare AMHT with BMH, we set the mini-batch size m'=200 and tuned the significance level $\epsilon = 0.01$ such that around 500 observations, which is 0.5% of the data, will be used at each iteration. Hence, such a run of AMHT cost more CPU time than BMH with $m = 200$. Note that each run of AMHT employed the same proposal distribution and consisted of the same number of iterations as BMH run. For each run of AMHT, we have also discarded the first 5000 iterations for the burn-in process and used the samples generated in the remaining 50,000 iterations for inference.

Table 4.3 compares the parameter estimates resulted from D&C , AMHT, and BMH (with k = 50, m = 200 and $\binom{n}{m}$-bootstrapping). BMH can produce very accurate estimates as much as D&C or AMHT, but is much faster than the other two methods although D&C and AMHT involve more observations at each iteration,

Table 4.3: Comparison of BMH with D&C and AMHT algorihtms for parameter estimation, where the numbers in upper row calculated by averaging estimates over 10 runs, and the number in the paranthesis is the standard deviation of the estimates. CPU(sec) is average running time in second.

Algorithm	β_0	β_1	β_2	β_3	$\log \sigma^2$	CPU(sec)
BMH	1.9997	0.2504	0.2463	0.0041	-1.3600	160.47
	(1.46e-4)	(1.34e-4)	(4.21e-4)	(5.90e-4)	(3.88e-4)	(11.16)
D&C	1.9997	0.2503	0.2491	0.0001	-1.3856	238.32
	(3.77e-5)	(3.40e-5)	(5.73e-5)	(1.07e-4)	(1.33e-4)	(10.50)
AMHT	1.9999	0.2500	0.2453	0.0050	-1.3716	303.81
	(5.47e-6)	(6.23e-6)	(1.40e-5)	(2.02e-5)	(2.93e-5)	(16.62)

and their standard deviation are smaller than BMH.

In Table 4.4, we report the MH, BMH (with $k = 50$, $m = 200$ and $\binom{n}{m}$-bootstrapping), AMHT (with $m' = 200$ and $\epsilon = 0.01$), and D&C estimates of $\sigma^2_{11}, \cdots, \sigma^2_{55}$, σ^2_{12} and σ^2_{34} based on the pooled samples from their respective runs, where σ^2_{ij} denotes the (i, j)th element of Σ_0. For the D&C method, Σ_0 was estimated by $\frac{n}{k}\hat{\Sigma}_{DC}$, where $\hat{\Sigma}_{DC}$ is the covariance matrix of the posterior samples pooled from 10 runs. For the BMH method, Σ_0 was estimated by $m\hat{\Sigma}_m$, where $m = 200$ and $\hat{\Sigma}_m$ is the covariance matrix of the posterior samples pooled from 10 runs. For AMHT method, Σ_0 was estimated by $n\hat{\Sigma}_{AMHT}$, where $\hat{\Sigma}_{AMHT}$ is the covariance matrix of the posterior samples pooled from 10 runs. For the MH method, Σ_0 was estimated by $n\hat{\Sigma}_n$, where $\hat{\Sigma}_n$ is the co-

Table 4.4: MH, BMH, D&C, and AMHT estimates of $\sigma^2_{11}, \cdots, \sigma^2_{55}$, σ^2_{12}, σ^2_{34}, and ρ_{β_2,β_3} obtained with pooled samples, where σ^2_{ij} denotes the (i, j)th element of Σ_0, and ρ_{β_2,β_3} denotes the correlation coefficient of β_2 and β_3

Method	σ^2_{11}	σ^2_{22}	σ^2_{33}	σ^2_{44}	σ^2_{55}	σ^2_{12}	σ^2_{34}	ρ_{β_2,β_3}
MH	0.2934	0.2661	1.4488	2.5658	1.9656	0.0580	-1.7657	-0.9158
BMH	0.2746	0.2718	1.6757	2.8428	2.0627	-0.0144	-1.9874	-0.9106
DNC	0.5035	0.5140	3.0750	5.3683	3.8867	0.0093	-3.7258	-0.9170
AMHT	0.3263	0.3788	2.0377	3.1848	2.7706	-0.0056	-2.3386	-0.9180

variance matrix of the posterior samples pooled from 10 runs. As in Table 4.2, the simulations have been thinned by a factor of 500, which ensures the pooled samples to be approximately mutually independent. Table 4.4 shows that BMH can produce correct estimates of Σ_0 using pooled samples, while D&C and AMHT cannot.

In summary, BMH can be very efficient for Bayesian analysis of big data, as it is able to incorporate all data information into a single run and thus inference can be made based on a single run. In contract, D&C needs to run for all subsets, otherwise the resulting inference can be severely biased.

4.2 BMH on Spatial Model

In this section, we will assess the performance of BMH algorithm on a spatial model. Consider the Gaussian geostatistical model,

$$\mathbf{Y} = \boldsymbol{\mu} + \mathbf{Z} + \epsilon, \quad \epsilon \overset{i.i.d}{\sim} N(0, \tau^2\mathbf{I}) \tag{4.1}$$

where $\mathbf{Y} = \{Y(s_1), \cdots, Y(s_n)\}^T$ denotes the observations at location $s_1, \cdots, s_n$, $\boldsymbol{\mu} = \{\mu(s_1), \cdots,\ \mu(s_n)\}^T$ denotes the mean vector of $\mathbf{Y}$, $\mathbf{Z} = \{Z(s_1), \cdots, Z(s_n)\}^T$ denotes a Gaussian process with mean, a vector of zeros, and covariance matrix $\boldsymbol{\Sigma} = \sigma^2\mathbf{R}$, where $\mathbf{R}$ is an exponential correlation function with elements of $\exp\{-\|s_i - s_j\|/\phi\}$ for $i, j = 1, \cdots, n$ where $\|\cdot\|$ is a distance measure. And τ^2 is the nugget variance. The model (4.1) can be extended to the regression setting with the mean $\mu(s_i)$ being replaced by

$$\mu(s_i) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j(s_i) \tag{4.2}$$

where $x_j(\cdot)$ denotes the jth explanatory variable, and β_j is the corresponding regrression coefficient. Under the model (4.1), $\mathbf{Y}$ follows a multivariate Gaussian distribution, $\mathbf{Y}|\boldsymbol{\theta} \sim N\left(\boldsymbol{\mu}, \sigma^2\mathbf{R} + \tau^2\mathbf{I}\right)$, and the log likelihood-like function of $D_i = (\mathbf{s}_i, \mathbf{Y}(\mathbf{s}_i))$

is defined as follows.

$$\log \tilde{f}(D_i|\boldsymbol{\theta}) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\sigma^2\mathbf{R}(\mathbf{s}_i) + \tau^2\mathbf{I}| \\ - \frac{1}{2}(\mathbf{Y}(\mathbf{s}_i) - \boldsymbol{\mu}(\mathbf{s}_i))^T \left(\sigma^2\mathbf{R}(\mathbf{s}_i) + \tau^2\mathbf{I}\right)^{-1}(\mathbf{Y}(\mathbf{s}_i) - \boldsymbol{\mu}(\mathbf{s}_i)) \qquad (4.3)$$

where $\mathbf{s}_i$ is subsample locations of corresponding subset D_i and $\mathbf{R}(\mathbf{s}_i) = \exp\{-d_i/\phi\}$ where d_i is a distance matrix between all locations in $\mathbf{s}_i$. We suggest following priors for the model (4.1) with exponential correlation function as non-informative priors.

$$\pi(\boldsymbol{\theta}) \propto \left(\frac{1}{\phi\sigma^2\tau^2}\right)^{m/n}$$

50 datasets are generated from the model (4.1) with uniformly distributed spatial sites of size n = 1000, 2000, 3000, 5000, and 10000 respectively. The covariate $\mathbf{x}$ is randomly generated from the normal distribution with mean zero and stadard deviation 0.5. The true parameters for the example are set as $\beta_0 = 1.0$, $\beta_1 = 1.0$, $\phi = 25.0$, $\sigma^2 = 1.0$, and $\tau^2 = 1.0$. We set subset size as $m = 100$ and 300 for each case of n, and the number of subsets is set as $k = 50$. The length of the Markov chain is 22,000, and the first 2,000-10,000 observations are discarded as burn-in period. The results of our example shows that BMH works very well for estimating the model (4.1). The resulting output is shown in Table 4.5. We took sample means of the chains as the estimators for the parameters. The numbers in Table 4.5 are the averages of the estimates from the 50 datasets, and the numbers in paranthesis are standard errors.

Note that, in this example, as n gets bigger, standard errors generally get smaller, and averages of the estimators get closer to the true values. When n is small ($n \leq 1000$), MLE is the fastest and the most accurate. However, when $n \geq 2000$, BMH

Table 4.5: Comparisons of BMH with MLE for 50 simulated datasets. n: size of dataset, m: size of subset, CPU(m): averaged CPU time(in minutes). The numbers in the parenthesis denote the standard error of the estimates.

n	m	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\phi}$	$\hat{\sigma}^2$	$\hat{\tau}^2$	$\hat{\phi}/\hat{\sigma}^2$	CPU(m)
1000	100	1.075	0.987	31.5	1.261	1.014	29.468	3.67
		(0.432)	(0.073)	(9.3)	(0.494)	(0.123)	(18.655)	(0.86)
	300	1.103	0.985	35.0	1.334	0.992	27.458	60.18
		(0.435)	(0.068)	(11.0)	(0.423)	(0.158)	(7.976)	(2.11)
	MLE	1.067	0.984	26.6	1.000	1.004	25.875	1.51
		(0.423)	(0.066)	(13.5)	(0.364)	(0.064)	(7.499)	(0.22)
2000	100	0.897	0.992	31.3	1.243	1.010	27.676	3.52
		(0.424)	(0.054)	(9.2)	(0.490)	(0.121)	(10.344)	(0.04)
	300	0.857	0.990	34.9	1.332	1.015	27.659	69.99
		(0.486)	(0.050)	(9.0)	(0.464)	(0.051)	(7.742)	(11.66)
	MLE	0.904	0.991	26.5	1.023	1.004	25.684	14.99
		(0.390)	(0.051)	(14.1)	(0.485)	(0.038)	(5.255)	(2.28)
3000	100	0.987	0.997	30.9	1.260	0.999	25.485	3.69
		(0.451)	(0.040)	(9.9)	(0.410)	(0.095)	(7.040)	(0.78)
	300	0.969	1.000	34.3	1.302	1.002	26.977	58.31
		(0.436)	(0.038)	(9.9)	(0.387)	(0.055)	(5.543)	(1.51)
	MLE	1.031	1.000	29.1	1.120	1.002	25.197	25.59
		(0.401)	(0.037)	(22.2)	(0.700)	(0.031)	(3.141)	(4.74)
5000	100	1.024	1.007	29.7	1.231	0.979	25.624	3.62
		(0.402)	(0.032)	(8.6)	(0.382)	(0.110)	(8.563)	(0.75)
	300	1.044	1.009	33.9	1.264	1.003	27.460	58.16
		(0.445)	(0.030)	(10.3)	(0.371)	(0.047)	(6.834)	(1.12)
	MLE	0.989	1.008	25.2	0.978	1.003	25.624	191.74
		(0.413)	(0.030)	(10.3)	(0.354)	(0.024)	(2.910)	(60.37)
10000	100	0.977	0.996	33.1	1.427	0.988	25.809	3.66
		(0.572)	(0.022)	(8.9)	(0.732)	(0.076)	(7.538)	(0.81)
	300	0.957	0.999	35.8	1.326	1.013	28.099	58.26
		(0.454)	(0.022)	(9.7)	(0.438)	(0.033)	(5.884)	(1.49)
	MLE	0.972	0.999	27.6	1.096	0.997	24.866	718.44
		(0.336)	(0.020)	(16.4)	(0.600)	(0.016)	(2.592)	(272.73)
True		1.000	1.000	25.0	1.000	1.000	25.0	-

with $m = 100$ gets 5 to 200 times faster than MLE. For the cases of BMH with $m = 300$, when $n \geq 5000$, BMH gets faster than MLE. It is because computational complexity of BMH, which is $O(m^3)$, is independent of n, whereas that of MLE is $O(n^3)$. Whatever the size of n is, the computation time of BMH is constant for fixed subsample size m. When the number of observation gets larger, it is getting hard to get MLE in matters of memory and speed, but there is still no problem in calculating BMH estimator. BMH could be more powerful for more complex

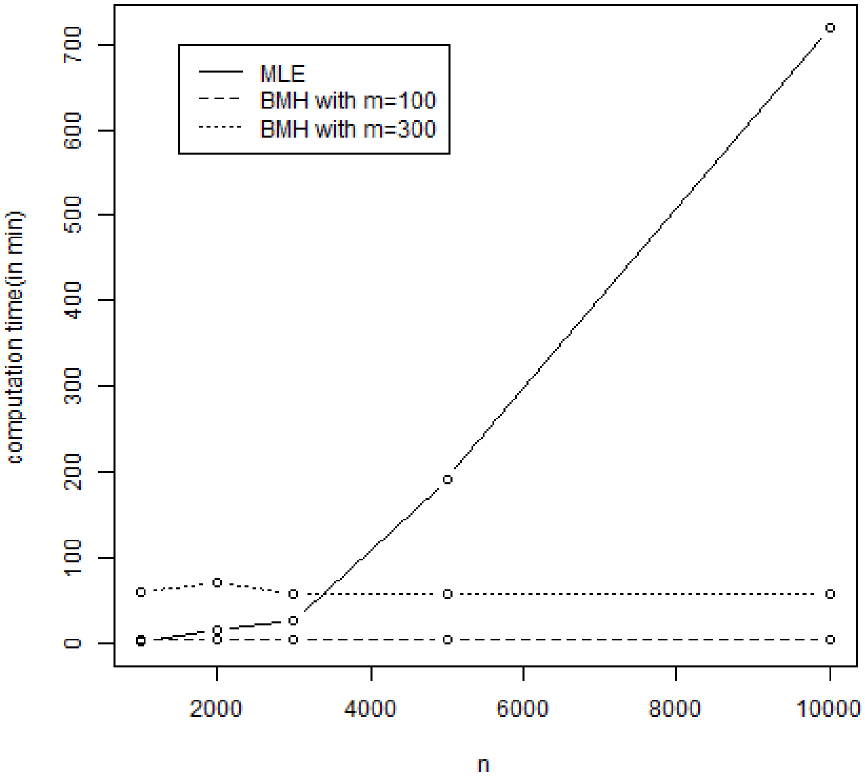

Figure 4.2: Speed of BMH and MLE with observation size of n: The solid line represents runing time of MLE in seconds, the dashed line represents the running time of BMH with $m = 100$, and the dotted line represents the running time of BMH with $m = 300$.

model such as spatio-temporal model. For T discrete time points, if we get n spatial observations, we need matrix multiplication of size n for T times including inversion of the matrix whose computational complexity is $O(Tn^3)$. If n is very large, computing MLE becomes even more serious problem than estimating spatial model. However, computational complexity of BMH in this case is $O(Tm^3)$, and it is still feasible even though we possiblely need to generate longer chain depending on the size of n. Figure 4.2 shows the average computational time in minutes for MLE, BMH with $m = 100$, and BMH with $m = 300$. The solid line represents runing time of MLE, the dashed line represents the running time of BMH with $m = 100$, and the dotted line represents the running time of BMH with $m = 300$. Because the subsample size m is fixed, running time of BMH is constant as the size of data, n, increases.

4.3 BMH on Spatio-Temporal Model

In this section, we will discuss the spatio-temporal model to apply for BMH algorithm. First we consider the modified AR model proposed by Sahu and Bakar (2012) to set spatio-temporal model. Let $\mathbf{Y}_t = \{Y(s_1, t), \cdots, Y(s_n, t)\}^T$ be the vector of observations and $\mathbf{Z}_t = \{Z(s_1, t)\ , \cdots,\ Z(s_n, t)\}^T$ be the vector of the true values in sites $\mathbf{s} = \{s_1, \cdots, s_n\}$ at time point t. The modified AR model is as follows:

$$\begin{aligned} \mathbf{Y}_t &= \boldsymbol{\mu} + \mathbf{Z}_t + \epsilon_t, \quad \epsilon_t \sim N(0, \tau^2\mathbf{I}), \\ \mathbf{Z}_t &= \rho\mathbf{Z}_{t-1} + \eta_t, \quad \eta_t \sim N(0, \Sigma_\eta), \\ \mathbf{Z}_0 &= \eta_0, \quad \eta_0 \sim N(0, \Sigma_\eta) \end{aligned} \tag{4.4}$$

where $\epsilon_t = (\epsilon(s_1, t), \cdots, \epsilon(s_n, t))$ is a vector of measurement errors with τ^2 as the nugget effect. $\eta = (\eta(s_1, t), \cdots, \eta(s_n, t)))$ is the spatially correlated error, and ρ is the autoregressive process parameter between sequential two time points. The covariance matrix $\Sigma_\eta = \sigma^2\mathbf{R}$ and $\mathbf{R}$ has elements $\exp\{-\|s_i - s_j\|/\phi\}$, $i, j = 1, \cdots, n$, where $\|\cdot\|$ is distance measure, and ϕ is range parameter of the exponential correlation function (Cressie, 1993). $\boldsymbol{\mu} = (\mu(s_1), \cdots, \mu(s_n))$ is a vector of mean function, which is constant throughout the all time points, and it can be expressed by linear function of possible covariates, $\mathbf{X}$, such that $\boldsymbol{\mu} = \mathbf{X}^T\beta$.

In the equation (4.4), the second stage model, which is correlation structure between $\mathbf{Z}_t$ and $\mathbf{Z}_{t-1}$, is AR(1) model, and, hence, we can write joint distribution of $\mathbf{Z}_t$ and $\mathbf{Z}_{t-1}$ as following.

$$\begin{pmatrix} \mathbf{Z}_t \\ \mathbf{Z}_{t-1} \end{pmatrix} \sim N(0, \Psi \otimes \Sigma_\eta), \qquad \Psi = \frac{1}{1-\rho^2}\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \tag{4.5}$$

where $\otimes$ is Kronecker product. For between time correlation matrix Ψ, see Brockwell and Davis (1991), p.81. Therefore, the joint distribution of $\mathbf{Y}_t$ and $\mathbf{Y}_{t-1}$ is as following.

$$\begin{pmatrix} \mathbf{Y}_t \\ \mathbf{Y}_{t-1} \end{pmatrix} \sim N(\boldsymbol{\mu}, \Psi \otimes \Sigma_\eta + \tau^2\mathbf{I}) \tag{4.6}$$

From the equation (4.6), we can derive the conditional distribution of $\mathbf{Y}_t|\mathbf{Y}_{t-1} \sim N(\boldsymbol{\mu}_Y, \boldsymbol{\Sigma}_Y)$

$$\begin{aligned}
\boldsymbol{\mu}_Y &= \boldsymbol{\mu} + \rho\Sigma_\eta W^{-1}(\mathbf{Y}_{t-1} - \boldsymbol{\mu}) \\
\boldsymbol{\Sigma}_Y &= \frac{1}{1-\rho^2}\left(W - \rho^2\Sigma_\eta W^{-1}\Sigma_\eta\right)
\end{aligned}$$

where $W = \Sigma_\eta + (1-\rho^2)\tau^2\mathbf{I}$. Ψ is the assymptotic correlation matrix for $t > c$ where c is large enough. Hence, the marginal distribution of $\mathbf{Y}_1$ is normal distriburion with mean $\boldsymbol{\mu}$ and covariance W_0 where $W_0 = \frac{1}{1-\rho^2}\Sigma_\eta + \tau^2\mathbf{I} = \frac{1}{1-\rho^2}W$.

By combining marginal distribution of $\mathbf{Y}_1$ and conditional distribution of $\mathbf{Y}_t|\mathbf{Y}_{t-1}$, the log likelihood-like function of D_i is definded as following.

$$\begin{aligned}
\log \tilde{f}(D_i|\boldsymbol{\theta}) &= \log \tilde{f}(\mathbf{Y}_1(\mathbf{s}_i)|\boldsymbol{\theta}) + \sum_{t=2}^{T} \log \tilde{f}(\mathbf{Y}_t(\mathbf{s}_i)|\mathbf{Y}_{t-1}(\mathbf{s}_i), \boldsymbol{\theta}) \\
&= -\frac{nT}{2}\log 2\pi + \frac{nT}{2}\log(1-\rho^2) - \frac{1}{2}\log|W_i| - \frac{T-1}{2}\log|K_i| \\
&\quad - \frac{1-\rho^2}{2}\left(\mathbf{Y}_1(\mathbf{s}_i) - \boldsymbol{\mu}(\mathbf{s}_i)\right)^T W_i^{-1}\left(\mathbf{Y}_1(\mathbf{s}_i) - \boldsymbol{\mu}(\mathbf{s}_i)\right) \\
&\quad - \frac{1-\rho^2}{2}\sum_{t=2}^{T}\big(\mathbf{Y}_t(\mathbf{s}_i) - \boldsymbol{\mu}(\mathbf{s}_i) - \rho\Sigma_i W_i^{-1}(\mathbf{Y}_{t-1}(\mathbf{s}_i) - \boldsymbol{\mu}(\mathbf{s}_i))\big)^T \\
&\quad \times K_i^{-1}\big(\mathbf{Y}_t(\mathbf{s}_i) - \boldsymbol{\mu}(\mathbf{s}_i) - \rho\Sigma_i W_i^{-1}(\mathbf{s}_i)(\mathbf{Y}_{t-1}(\mathbf{s}_i) - \boldsymbol{\mu}(\mathbf{s}_i))\big) \qquad (4.7)
\end{aligned}$$

where $\Sigma_i = \sigma^2\mathbf{R}(\mathbf{s}_i)$, $W_i = \Sigma_i + (1-\rho^2)\tau^2\mathbf{I}$, and $K_i = W_i - \rho^2\Sigma_i W_i^{-1}\Sigma_i$. To apply BMH on the spatio-temporal model, we also define non-informative priors, $\pi(\boldsymbol{\theta})$, as following.

$$\pi(\boldsymbol{\theta}) \propto \left(\frac{1}{\phi\sigma^2\tau^2}\right)^{m/n}$$

To apply BMH to the examples, we reparameterize ϕ, σ^2, τ^2, and ρ in their logarithms so that their parameter spaces are always positive. In this section, 50 simulation datasets are generated by the model (4.4) using the package `geoR`. At $n = 2000$ spatial sites within bounded region on $[0, 100] \times [0, 100]$ are randomly selected for $T = 120$ descrete time points. To assess the performance of BMH according to the between time autocorrelation, ρ, we set different values for ρ. We assume no covariates in the model, so the mean function is constant, β. The true values set on each parameter are set as following.

$$\beta = 5.0, \phi = 25.0, \sigma^2 = 1.0, \tau^2 = 1.0, \rho = \begin{cases} 0.2 \\ 0.7 \end{cases}$$

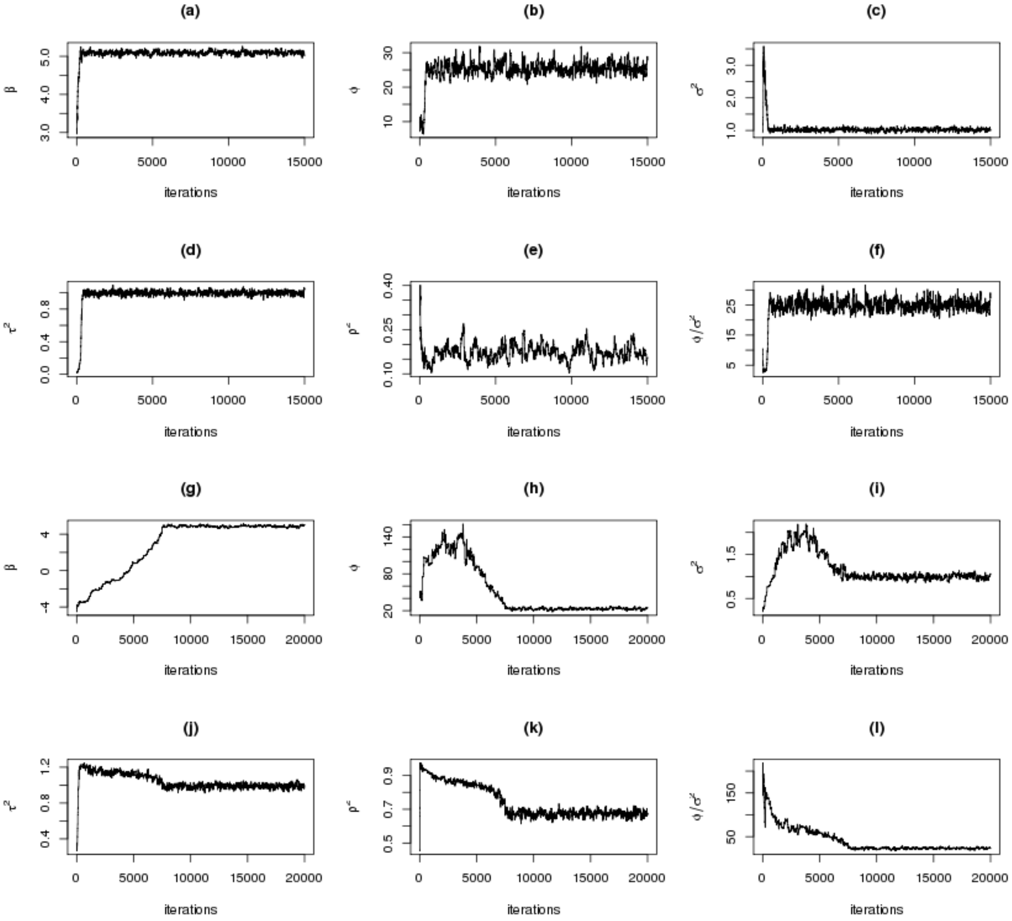

Figure 4.3: Trace plot of samples of the parameters from the posterior distribution by BMH algorithm: (a)-(f) are trace plots where $\rho = 0.2$, and (g)-(l) are trace plots where $\rho = 0.7$. (a)-(f) and (g)-(l) represent β, ϕ, σ^2, τ^2, and ρ respectively.

Through out the paper, we used $k = 50$ parallel threads on a cluster machine which uses Quad-Core AMD Opteron$^{\text{TM}}$ Processor 8382 @2.6Ghz. For the case when $\rho = 0.2$, each run is consisted of 15,000 iterations, and the first 5,000 iterations were discarded for burn-in process whereas runs for the case of higer autocorrelation, $\rho = 0.7$ are consisted of 20,000 iterations. The first 10,000 iterations were discarded for the burn-in process.

Figure 4.3 shows the trace plot of the samples generated from the posterior distribution for each case of ρ. When ρ is relatively high, the logner burn-in process is needed. The subsample sizes are set to be $m = 100$, and the stepsize is set by 0.2 to have acceptance rate of $0.15 \sim 0.23$. Table 4.6 summarizes the results of the BMH runs, the averages of 50 estimates of BMH and MLE, and numbers in the parenthesis denote the standard errors of the estimates. Notice that BMH is more than 30 times faster than MLE for estimating spatio-temporal model. The averages of BMH estimators are very close to MLE which is unbiased.

4.3.1 Comparison Study with Existing Methods

In this section, we illustrate the performance of BMH by comparing other methods, AMHT and D&C with spatio-temporal model example. Under the same parameter setting with previous example of this section, 10 data sets with $n = 10,000$ samples are generated for $T = 120$ descrete time points. Again we set $k = 50$, and subsample size $m = 200$ for BMH and AMHT, and D&C also uses n/k =200 observations in each groups. At for this example, temporal subsets are used for BMH method. At every iteration of BMH, m subsamples are selected as a subset of Bootstrap sample, and then, $T_s = 50$ time points are also selected among 120

Table 4.6: BMH result for the spatial-temporal model with nugget effect: The first column, ρ represents the true values for between time autocorrelation coefficient.

ρ	Method	$\hat{\beta}$	$\hat{\phi}$	$\hat{\sigma}^2$	$\widehat{\tau^2}$	$\hat{\rho}$	$\widehat{\phi/\sigma^2}$	CPU(min)
0.2	BMH	5.001 (0.046)	25.14 (1.426)	1.011 (0.036)	0.999 (0.007)	0.195 (0.017)	24.90 (1.005)	7.13 (0.7)
	MLE	4.9985 (0.046)	25.07 (1.098)	1.0033 (0.036)	1.0004 (0.004)	0.1985 (0.010)	24.99 (0.479)	311.13 (101.8)
0.7	BMH	5.012 (0.104)	25.29 (1.355)	1.003 (0.036)	1.002 (0.007)	0.699 (0.010)	25.27 (0.976)	9.08 (0.1)
	MLE	5.008 (0.109)	25.02 (1.030)	1.001 (0.034)	1.000 (0.004)	0.699 (0.005)	25.01 (0.413)	281.54 (78.6)

Table 4.7: Results for the spatial-temporal model of BMH, AMHT, and D&C with $m = 200$ and $k = 50$. From the left column, ϕ represents the true values of range parameter, ρ represents the true values of between time autocorrelation coefficient. CPU(m) is running time in minute.

ϕ	ρ	Method	$\hat{\beta}$	$\hat{\phi}$	$\hat{\sigma}^2$	$\hat{\tau^2}$	$\hat{\rho}$	CPU(m)
25	0.2	BMH	5.009 (2.9e-3)	25.595 (6.8e-2)	1.016 (2.6e-3)	1.001 (5.3e-4)	0.188 (8.1e-2)	61.3 (9.2)
		AMHT	5.013 (2.3e-3)	25.773 (5.9e-2)	1.021 (2.8e-3)	1.001 (4.8e-4)	0.187 (6.1e-4)	210.4 (4.6)
		D&C	5.010 (2.4e-3)	25.314 (6.7e-2)	1.013 (2.4e-3)	0.999 (4.1e-4)	0.193 (5.2e-4)	98.3 (1.4)
	0.7	BMH	5.009 (8.4e-3)	25.582 (5.6e-2)	1.013 (3.3e-3)	1.000 (5.0e-4)	0.703 (5.8e-4)	69.0 (8.5)
		AMHT	5.016 (6.9e-3)	25.837 (5.1e-2)	1.021 (3.3e-3)	1.002 (4.4e-4)	0.702 (5.3e-4)	161.3 (1.7)
		D&C	5.001 (7.6e-3)	25.399 (6.7e-2)	1.013 (3.0e-3)	0.999 (4.5e-4)	0.699 (4.9e-4)	105.3 (4.9)
75	0.2	BMH	5.002 (3.6e-3)	79.246 (2.6e-1)	1.012 (4.3e-3)	1.003 (2.1e-4)	0.181 (1.8e-3)	64.8 (6.6)
		AMHT	5.000 (3.6e-3)	80.215 (1.7e-1)	1.026 (3.5e-3)	1.003 (2.2e-4)	0.181 (1.4e-3)	192.4 (3.0)
		D&C	5.011 (2.8e-3)	76.884 (2.4e-1)	0.993 (4.7e-3)	1.002 (2.2e-4)	0.192 (1.5e-3)	103.1 (3.1)
	0.7	BMH	4.978 (1.2e-2)	84.409 (5.1e-1)	1.077 (3.6e-3)	1.003 (4.0e-4)	0.707 (8.3e-4)	63.4 (5.2)
		AMHT	4.982 (9.8e-3)	86.510 (5.0e-1)	1.101 (4.0e-3)	1.004 (3.6e-4)	0.705 (7.7e-4)	170.7 (3.0)
		D&C	5.038 (9.4e-3)	80.581 (3.0e-1)	1.055 (2.9e-3)	1.001 (3.1e-4)	0.701 (7.8e-4)	103.0 (2.5)

descrete time points in the subsampled chunck. Temporal subset should be consecutive to detect temporal correlation, and it can be contructed by following. First, generate random number t_0 from $Unif(1, T - T_s)$, and then, select time points of $[t_0, t_0 + 1, \cdots, t_0 + T_s - 1]$ among the previously selected random subset. By using this scheme, we can run BMH more faster. However, this temporal subsetting should be carfully considered because if we take too small T_s to estimate temporal correlation, location parameter and nugget parameter cannot be correctly estimated, and it affects on the estimate of range parameter. The step size is set to by 0.15 for all three methods to have acceptance rate of $0.15 \sim 0.30$. Significance level, is set by $\epsilon = 0.02$ for AMHT. 55,000 samples are generated from the posterior distribution, and the first 5,000 samples are discarded as burn-in process. From remaining 50,000 samples, 500 samples are selected systemically as one in every 100 points to have *i.i.d.* random numbers. Table 4.7 summarizes the result of the BMH, AMHT, and D&C runs, the averages of 10 estimates of the methods.

In this example, the same number of observations are used at each iteration of BMH and D&C, and the same number of mini-batch, m' is used for AMHT. However, D&C and AMHT used $T = 120$ of whole time period whereas BMH used $T_s = 50$ of subset period. Hence, BMH cost less memory and computation time than the other two methods. In the Table 4.7, obivously, standard errors for the parameter estimates of D&C and AMHT are a little smaller than that of BMH because their number of samples actually used are bigger, and the estimates of D&C are slightly more accurate than that of BMH, but still BMH estimates are quite usible as much as D&C, and seem even better than that of AMHT.

5. REAL DATA ANALYSIS

5.1 US Precipitaion

In this section, we assess the performance of BMH using Spatial data. The data we used in this section is the US precipitation data from *National Climatic Data Center for years 1895 to 1997*, which are available at *www.image.ucar.edu/GSP/Data/US.monthly.met/*. The reason that we used these data is that they are fully observed for all 103 years with missing data imputed by Johns et al. (2003), and are very large so that they can be good examples for assessing the performance of BMH on big data. The observed spatial sites for precipitation is $n = 11918$. US precipitation data is seriously right skewed, and can not be applied to the gaussian model. To have symmetric distribution, we used anomalies that is standardized square root of the monthly precipitations. Hence, the mean function in our model is assumed to be constant, which is zero throughout all the sites.

Among the 103 years of US precipitation data, April 1948 is selected to assess performance of BMH on spatial model. First, the dataset is randomly divided into

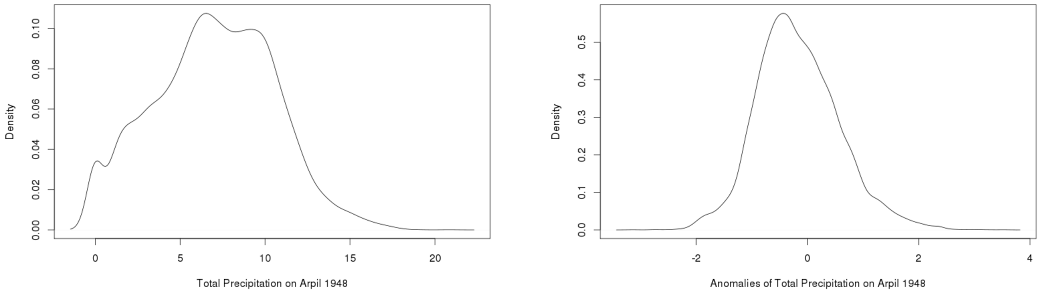

Figure 5.1: Total precipitation(left) and Anomalies(right) in April 1948

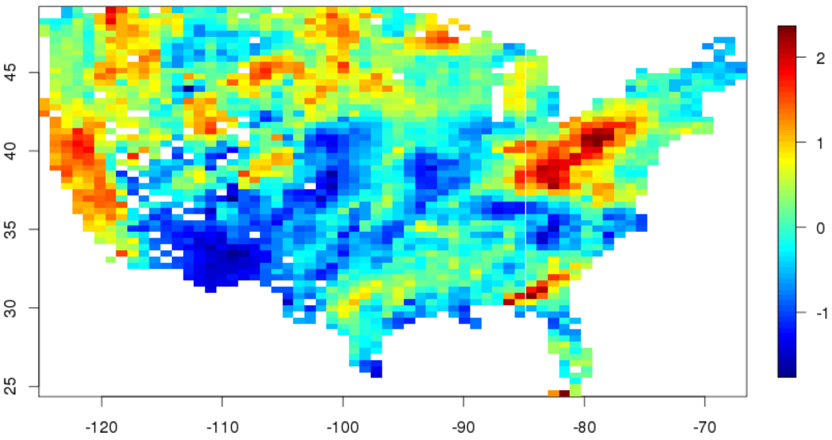

Figure 5.2: Contourplot of April 1948 Anomalies of US precipitation

$n = 11000$ of training set and $n = 918$ of test set. BMH with $m = 100$ and 300 were applied to the trainning set for five times. Corresponding stepsizes are set to be 1.5 and 1.0 to have acceptance rate of $0.15 \sim 0.20$. The total length of chain is 15,000, and the first 5,000 iterations are discarded as burn-in process. Figure 5.3 shows the trace plots of BMH samples. The black lines represent BMH chains when $m = 100$, and the red lines represent BMH chains when $m = 300$. The chains when $m = 300$ have much less variation than when $m = 100$. Table 5.1 shows averages of the 5

Table 5.1: Parameter estimation for April 1948 US precipitation

Method	m	β	ϕ	σ^2	τ^2	ϕ/σ^2	CPU(min)
BMH	100	0.0943	318.48	0.8729	0.0491	352.02	3.39
		(0.0247)	(37.73)	(0.0569)	(0.0096)	(19.59)	(0.30)
	300	0.0500	218.47	0.7328	0.0274	293.04	57.73
		(0.0113)	(26.23)	(0.0481)	(0.0041)	(10.89)	(2.74)
MLE		-0.2348	167.86	0.8625	0.0270	194.62	32331.22

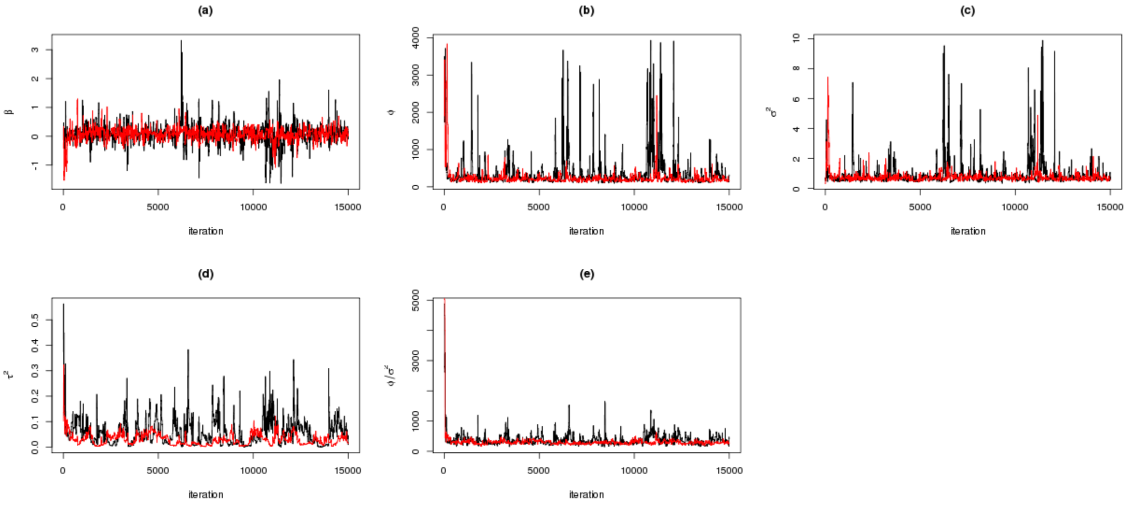

Figure 5.3: Trace plot of the parameters in spatial model by BMH algorithm for April 1984 US precipitation: (a)-(e) represent β, ϕ, σ^2, τ^2, and ϕ/σ^2 respectively. The black line is for $m = 100$, and the red line is for $m = 300$.

estimates and their stadard errors. As subset size m gets bigger, stadard errors of all parameters are reduced.

5.1.1 Kriging

Under the model (4.1), a joint distribution of the dependent variable is multivariate normal distribution for fixed domain. Hence, the joint distribution at observed locations and new locations given parameters is as follows.

$$\left(\begin{matrix} \mathbf{X}_{\text{obs}} \\ \mathbf{X}_{\text{new}} \end{matrix} \middle| \, \boldsymbol{\theta} \right) \sim N \left(\begin{pmatrix} \boldsymbol{\mu}_{\text{obs}} \\ \boldsymbol{\mu}_{\text{new}} \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix} \right) \tag{5.1}$$

where $\boldsymbol{\theta}$ is a set of all parameters in the model, $\boldsymbol{\Sigma}_{11}$ is a covariance matrix of observed locations, $\boldsymbol{\Sigma}_{12} = \boldsymbol{\Sigma}_{21}^T$ is a covariance matrix between observed locations and new locations, and $\boldsymbol{\Sigma}_{22}$ is a covariance matrix of new locations. With fixed parameters, predictors for the new locations are assumed to follow conditional normal distribution

(Handcock and Stein, 1993). And the conditional distribution of $\mathbf{X}_{\text{new}}|\mathbf{X}_{\text{obs}}, \boldsymbol{\theta}$ is,

$$\mathbf{X}_{\text{new}}|\mathbf{X}_{\text{obs}}, \boldsymbol{\theta} \sim N\left(\boldsymbol{\mu}_{\text{new}} + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\left(\mathbf{X}_{\text{obs}} - \boldsymbol{\mu}_{\text{obs}}\right), \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}\right) \tag{5.2}$$

Here, we still need to calculate inversion of covariance matrix for observed locations, $\boldsymbol{\Sigma}_{11}^{-1}$, and it is infeasible if the number of observation is very large. To avoid this problem, Cressie (1993) suggested to use only neighborhoods that will typically have more substantial weights. We will call this method Local Kriging, and the Local Kriging is used for prediction throughout this paper. Let s_0 be the location that we need to predict, and let $\mathbf{s}_1$ be the neighborhoods near s_0. Then, we can define covariances with respect to s_0 and $\mathbf{s}_1$, $cov(\mathbf{s}_1) = \Sigma_1$ and $cov(\mathbf{s}_1, s_0) = \Sigma_{10}$, and the Local Kriging estimator $\hat{X}(s_0)$ is defined as following.

$$\hat{X}(s_0) = \mu(s_0) + \Sigma_{01}\left(\Sigma_1 + \tau^2\mathbf{I}\right)^{-1}\left(\mathbf{X}(\mathbf{s}_1) - \boldsymbol{\mu}(\mathbf{s}_1)\right)$$

where $\boldsymbol{\mu}(\mathbf{s})$ is mean function at location $\mathbf{s}$, and $\mathbf{X}(\mathbf{s}_1)$ is observations at location set $\mathbf{s}_1$, which is the neighborhoods near s_0. In the Bayesian context, predictor follows posterior predictive distribution, and hence, it is defined by

$$f\left(X(s_0)|\mathbf{X}(\mathbf{s}_1)\right) = \int f(X(s_0)|\mathbf{X}(\mathbf{s}_1), \boldsymbol{\theta})\pi(\boldsymbol{\theta}|\mathbf{X}(\mathbf{s}_1))d\boldsymbol{\theta} \tag{5.3}$$

where $f(\cdot)$ is a density function of conditional normal distribution defined in (5.2). The point estimation of the predictor can be earned by calculating expected value of the posterior predictive distribution, (5.3).

For the prediction of US precipitation, neighborhood distance δ is set as 40, 50, 100 and 150. MSPEs are calcluated for 5 times for each setting of δ, with MLE and BMH estimators by $m = 100$ and $m = 300$. The resulting output is shown in

Table 5.2. The numbers in Table 5.2 are the averages of MSPEs and the numbers in parenthesis are standard errors. In this paper, MSPE is defined by following.

$$\text{MSPE} = E\left(\hat{y}_i - y_i\right)^2 \approx \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \left(\hat{y}_i - y_i\right)^2$$

where y_i is the i-th observation of test dataset, $\hat{y}_i$ is predictor for the test dataset, and n_{test} is the number of observations in the test dataset, which is 918 in this example. The average numbers of observations for $\delta = 40, 50, 100$, and 150 are about 25, 38, 135, and 277 respectivly. Thining is made for calculating predictive posterior distribution. Among 10,000 samples in the chain after the burn-in process, samples are selected in every 100 samples. Hence, total of 100 samples are applied to calculate the predictive posterior distribution. The prediction results in Table 5.2 indicate that neighborhood distance δ should be at least 50 mile to provide sufficiently precise prediction for this dataset. When we set $\delta = 40$, the median number of neithborhoods is 20, and such points that have small number of neighborhoods would not be able to be correctly predicted. So, choice of δ can be done by checking the number of neighborhoods, and finding δ having sufficient number of neighborhoods. However, the speed of Local Kriging depends on the number of neighborhoods as whose computational

Table 5.2: Averages of MSPEs using neighborhoods within distances δ. Numbers in paremthesis are stadard errors of the MSPEs. CPU(sec) represents running time in seconds for calculating single MSPE.

	Neighborhood distance(δ)			
m	40	50	100	150
100	0.06891(2.0e-3)	0.06818(2.0e-3)	0.06830(2.2e-3)	0.06827(2.2e-3)
300	0.06744(4.2e-4)	0.06663(3.7e-4)	0.06664(3.9e-4)	0.06660(3.9e-4)
MLE	0.06681	0.06617	0.06619	0.06615
CPU(sec)	6.62(0.03)	11.33(0.03)	149.21(4.20)	1275.93(5.70)

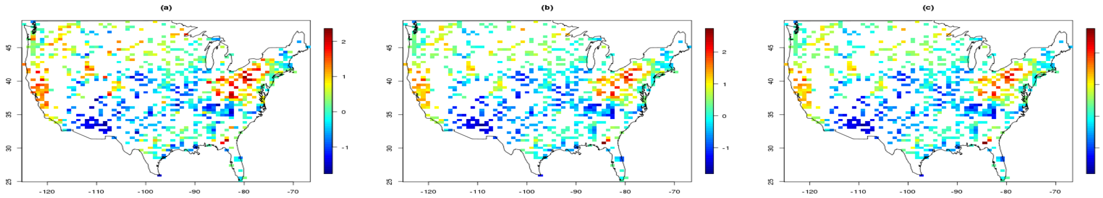

Figure 5.4: Observed and predicted precipitation for April 1948: (a) is the true values in test dataset, (b) is predicted values of Local Kriging of $\delta = 50$ with BMH estimator $m = 300$, and (c) is predicted values of Local Kriging of $\delta = 50$ with MLE.

complexity is $O(n_{\text{test}} \times n_b^3)$ where n_b is number of neighborhoods. Hence, considering both of minimum and maximum number of neighborhoods is needed. We suggest at least 5 and at most 200 neighborhoods as reasonable number of neighborhoods. The Local Kriging with parameters estimated by BMH with $m = 300$ provides enough precise prediction as much as that by MLE, but notice that BMH gurantees highly faster estimation.

5.1.2 Comparison Study

To compare with BMH, the data is applied to AMHT and D&C for five times. For AMHT mini-batch size is set by $m' = 100$ and significance level is set by $\epsilon = 0.002$, for D&C, $k = 50$ groups are used, and so in each group $n/k = 220$ observations are used. Table 4.6 shows estimates from the three methods. Numbers in upper rows are averages of the five parameter estimates, and the numbers in paranthesis in the lower rows are standard errors of the average estimates. We set parameter space for the range parameter ϕ as $[0, 3000]$ because a distance bwteen the east cost to the west cost of US is approximated 3,000 miles and ϕ should be smaller than 3,000 miles. 15,000 samples were generated and the first 5,000 samples were discarded as burn-in process as we did for BMH algorithm. This data is originaly imputed by Johns et al. (2003), and he didn't set the nugget effect on his model. Interestingly, AMHT

Table 5.3: Parameter estimation of sptial model for April 1948 Anomalies of US precipiation.

Method	μ	ϕ	σ^2	τ^2	ϕ/σ^2
BMH	0.0943	318.48	0.8729	0.0491	352.02
	(0.0247)	(37.73)	(0.0569)	(0.0096)	(19.59)
AMHT	-0.2058	309.18	0.7958	0.0000	384.81
	(0.0036)	(3.99)	(0.0089)	(0.0000)	(0.5673)
D&C	0.1022	324.52	0.9100	0.0029	361.49
	(0.0192)	(21.47)	(0.0117)	(0.0053)	(10.25)

estimated the nugget effect, $\tau^2 = 0$ which is true for this data. Certainly, AMHT is good for point estimation, but its poor performance on the covariance estimation of posterior distribution is not suitable for performing bayesian inference.

Prediction were made by parameter samples generated from the posterior distribution using BMH and AMHT, and also we did kriging by simply pluging in the parameter estimates from D&C method because D&C has different chains for each partition and it is hard to define monte carlo integration using different chains. The neighborhood size was set by $\delta = 50$, and MSPEs for the prediction using the results of AMHT and D&C are 0.07173 and 0.07917 respectively. This is bigger than that of BMH, which is 0.06818 with same size of neighborhood 50. Prediction of BMH was better than that of other methods because BMH samples are well-described its posterior distribution. When we calculate predictive posterior distribution, $\pi(y^*|\mathcal{D}) = \int_\Theta P(y^*|\mathcal{D}, \boldsymbol{\theta})\pi(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}$, we need to do monte carlo integration with respect to parameters generated from posterior distribution, where y^* is the value we need to predict. And as the samples correctly descibe population posterior, this integration will be converged well to its true value.

6. CONCLUSION

In this paper, we have proposed the BMH algorithm as a basic MCMC algorithm for Bayesian analysis of big data. The BMH algorithm is workable on parallel and distributed architectures and avoids repeated scans of the full dataset in iterations, and is thus feasible for big data problems. Compared to the popular divide-and-conquer strategy, BMH is generally more efficient as it can asymptotically integrate the whole data information into a single simulation run. The BMH algorithm is very flexible. Like the Metropolis-Hastings algorithm, it can serve as a basic building block for developing advanced MCMC algorithms that are feasible for big data problems. Compared to the existing big data analysis methods, such as aggregated estimating equation, resampling-based stochastic approximation, bag of little bootstraps, and approximate MH test, a unique power of BMH is that it tames the powerful MCMC methods to be used for big data analysis, such as parameter estimation, optimization and model selection. BMH provides a simple yet effective way of uncertainty quantication for big data problems.

Let T denote the number of iterations of BMH. Then the overall computational complexity of BMH can be expressed as $O(mkT)$, which is the same for both resampling schemes. Note that the computational complexity of BMH does not directly depend on n, although m, k and T can all increase with n. As shown in Chapter 3, for BMH, we can set $m = O(n^{\gamma})$ and $k = O(n^{\gamma+\epsilon_0})$ for $\gamma < 1/2$ and any $\epsilon_0 > 0$. Hence, the overall computational complexity of BMH is $O(n^{2\gamma+\epsilon_0}T)$. In a parallel implementation, the time complexity of BMH is $O(n^{\gamma}T)$. For a parallel implementation of the MH algorithm on the full data, the time complexity is $O(n^{1-\gamma-\epsilon_0}T)$ if $n^{\gamma+\epsilon_0}$ nodes are used. Since γ usually takes a very small value and ϵ_0 is nearly zero, BMH can be much faster than MH.

BMH aims to provide a numerical approximation to the full data posterior for big data problems. The BMH approximation preserves all features of the full data posterior, such as marginal and correlation structures, which can be inferred from its samples. As shown in Table 4.2, the approximation can be rather accurate (up to a known scale factor). The BMH algorithm proposes to replace the full data log-likelihood by a Monte Carlo average of the log-likelihoods that are calculated in parallel from multiple bootstrap samples. As an alternative strategy, one may try to replace the likelihood function by its Monte Carlo average in simulations. Although, in theory, this averaged likelihood method may go through under suitable conditions, our numerical results show that it can be much less efficient than BMH in terms of accuracy of the resulting parameter estimates. Compared to the averaged likelihood method, BMH has some significant advantages. As explained previously, it follows from Jensens inequality that for any $m \in N$, the mode of $g_m(\mathcal{D}|\boldsymbol{\theta})$ is identical to $\boldsymbol{\theta}^*$; that is, the maximum mean log-likelihood estimator is identical to the true parameter. This makes $l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\theta})$ perform like a regular log-likelihood function when n and k are large. However, this property does not hold for the maximum mean likelihood estimator. We note that the maximum mean log-likelihood estimator has been explored in Liang et al. (2013) using a resampling-based stochastic approximation method in the context of large geostatistical data. In this paper, we have considered only the use of BMH in parameter estimation. Applying BMH to model selection is straightforward. For example, BMH can be combined with the reversible jump MCMC algorithm (Green, 1995) in a similar way to tempering BMH for tackling the problem of model selection. In addition to the parameter estimation and model selection problems, BMH can also be applied to optimization problems by running it under the framework of simulated annealing (Kirkpatrick et al., 1983). This leads to the annealing BMH algorithm. A further exploration for the performance of these algorithms is of great interest.

REFERENCES

Adrieu, C. and Robert, G. (2009), “The Pseudo-Marginal Approach for Eddicient Monte Carlo Computations,” *Annals of Statistics*, 37, 697–725.

Brockwell, P. J. and Davis, R. A. (1991), *Time Series: Theory and Methods*, New York: Springer-Verlag, 2nd ed.

Chen, C. (1985), “On Asymptotic of Limiting Density Functions with Bayesian Implications,” *Journal of the Royal Statistical Society, Series B*, 47, 540–546.

Chen, M. H. and Schmeiser, B. W. (1996), “General Hit-and-Run Monte Carlo Sampling for Evaluating Multidimensional Integrals,” *Operational Research Letters*, 15, 161–169.

Cressie, N. A. C. (1993), *Statistics for Spatial Data*, New York: Wiley, 2nd ed.

Dawid, A. (1970), “On the Limiting Normality of Posterior Distributions,” *Mathematical Proceedings of the Cambridge Philosophical Society*, 67, 625–633.

Dekkers, A. and Aarts, E. (1991), “Global Optimization and Simulated Annealing,” *Mathematical Programming*, 50, 367–393.

Gelman, A., Roberts, G., and Gilks, W. (1996), “Efficient Metropolis Jumping Rules,” in *Bayesian Statistics*, eds. Bernado, J. M. et al., OUP, vol. 5, p. 599.

Green, P. (1995), “Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination,” *Biometroka*, 82, 711–732.

Handcock, M. S. and Stein, M. L. (1993), “A Bayesian Analysis of Kriging,” *Technometrics*, 35, 403–410.

Heyde, C. and Johnstone, I. (1979), “On Asymptotic Posterior Normality for Stochastic Processes,” *Journal of Royal Statistical Sociery, Series B*, 41, 184–189.

Hoeffding, W. (1948), “A Nonparametric Test for Independence,” *The Annals of*

Mathematical Statistics, 19.

Johns, C. J., Nychka, D., Kittel, T. G. F., and Daly, C. (2003), "Infilling Sparse Records of Spatial Fields," *Journal of the American Statistical Association*, 98, 796–806.

Kass, R., Tierney, L., and Kadane, J. (1990), "The Validity of Posterior Expansions Based on Laplace's Method," in *Essays in Honor of George Barnard*, eds. Gsisser, S., Hodges, J., Press, S., and Zellner, A., Amsterdam: North-Holland, pp. 473–488.

Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983), "Optimization by Simulated Annealing," *Science*, 220, 671–680.

Kleiner, A., Talwalkar, A., Sarkar, P., and Jordan, M. I. (2014), "A Scalable Bootstrap for Massive Data," *Journal of Royal Statistical Society series B*, 76, 795–816.

Korattikra, A., Chen, Y., and Welling, M. (2014), "Austerity in MCMC Land: Cutting the Metropolis-Hastings Budget," *In Proceeding of the International Conference on Machine Learning(ICML)*, 2014, See also arXiv:1304.5299v4.

Laney, D. (2012), "The Importance of 'Big Data': A Definition," *Gartner*, Retrieved 21 June 2012.

Lee, A. J. (1990), *U-Statistics: Theory and Practice*, New York: CRC Press.

Liang, F., Cheng, Y., Song, Q., Park, J., and Yang, P. (2013), "A Resampling-Based Stochastic Approximation Method for Analysis of Large Geostatistical Data," *Journal of the American Statistical Association*, 108, 325–339.

Liang, F. and Jin, I. H. (2013), "A Monte Carlo Metropolis-Hastings Algorithm for Sampling from Distributions with Intractable Normalizing Constants," *Neural Computation*, 25, 2199–2234.

Lin, N. and Xi, R. (2011), "Agrregated Estimating Equation Estimation," *Statistics and Its inference*, 4, 73–83.

Lindley, D. (1961), "The Use of Prior Probability Distributions in Statistical Inference and Decisions," in *In Proceedings of the Fourth Berkeley Symposium*, Berkeley, CA: University of California Press, vol. 1, pp. 473–488.

— (1980), "Approximate Bayesian Methods," in *Bayesian Statistics*, eds. Bernardo, J., Degroot, M., Lindley, D., and Smith, A., Valencia, Spain: Univertiry Press, pp. 223–245.

Miyata, Y. (2004), "Fully Exponential Laplace Approximations using Asymptotic Models," *Journal of the American Statistical Association*, 99, 1037–1049.

Sahu, S. K. and Bakar, K. S. (2012), "Hierarchical Bayesian Auto Regressive Models for Large Space-Time Data with Applications to Ozone concentration modeling," *Applied Stochatic Models in Bussiness and Industry*, 28, 395–415.

Stephens, M. (2000), "Dealing with Label Switching in Mixture Models," *Journal of the Royal Statistical Society, Series B*, 62, 795–809.

Tierney, L. (1994), "Markov Chains for Exploring Posterior Distributions," *The Annals of Statistics*, 22, 1701–1762.

Walker, A. (1969), "On the Asymptotic Behavior of Posterior Distributions," *Journal of the Royal Statistical Society, Series B*, 31, 80–88.

APPENDIX A

Proof of Lemma 3.3.2 Let

$$Q(\boldsymbol{\theta}, \psi(\lambda_{m,n,k}(\boldsymbol{\theta}, \boldsymbol{D_s}, \boldsymbol{\vartheta}) > \epsilon)) = \int_{\left\{(\boldsymbol{\vartheta}, \boldsymbol{D_s}) : \lambda_{m,n,k}(\boldsymbol{\theta}, \boldsymbol{D_s}, \boldsymbol{\vartheta}) > \epsilon\right\}} \psi(d\boldsymbol{D_s}) Q(\boldsymbol{\theta}, d\boldsymbol{\vartheta}),$$

where $\boldsymbol{D_s}$ is treated as a continuous variable for the notational simplicity. It follows from (3.12) and condition (B) that for any $\boldsymbol{\theta} \in \Theta$ and any $\epsilon > 0$,

$$\lim_{k\to\infty} \lim_{n\to\infty} Q(\boldsymbol{\theta}, \psi(\lambda_{m,n,k}(\boldsymbol{\theta}, \boldsymbol{D_s}, \boldsymbol{\vartheta}) > \epsilon)) = 0.$$

Then the remaining part of the proof follows the proof of Lemma 2 of Liang and Jin (2013). This completes the proof of Lemma 3.3.2.

Proof of Equation (3.16) It follows from (3.15) and the telescoping sum decomposition formula that

$$\begin{aligned}
\|\tilde{P}_{m,n,k}\phi(\boldsymbol{\theta}) - P_m^{\kappa_0}\phi(\boldsymbol{\theta})\| &\leq \sum_{\boldsymbol{D_s}\in\mathbb{D}} \|P_{m,n,k,\boldsymbol{D_s}}^{\kappa_0}\phi(\boldsymbol{\theta}) - P_m^{\kappa_0}\phi(\boldsymbol{\theta})\|\psi(\boldsymbol{D_s}) \\
&= \sum_{\boldsymbol{D_s}\in\mathbb{D}} \|\sum_{l=0}^{\kappa_0-1} P_m^l (P_{m,n,k,\boldsymbol{D_s}} - P_m) P_{m,n,k,\boldsymbol{D_s}}^{\kappa_0-(l+1)}\phi(\boldsymbol{\theta})\|\psi(\boldsymbol{D_s}) \\
&\leq \kappa_0 \sum_{\boldsymbol{D_s}\in\mathbb{D}} \|P_{m,n,k,\boldsymbol{D_s}}\phi(\boldsymbol{\theta}) - P_m\phi(\boldsymbol{\theta})\|\psi(\boldsymbol{D_s}).
\end{aligned}$$

Then, following the same reasoning as in the proof of Lemma 3.3.2, we have (3.16) holds.

Proof of Equation (3.17) Let $\mu = E[\log f(X|\boldsymbol{\theta})]$, let $z_i = \log f(X_i|\boldsymbol{\theta}) - \mu$ for $i = 1, \cdots, n$, let $\sigma^2 = Var(z_i)$, and let $\{y_1, \cdots, y_{mk}\}$ denote mk samples drawn without replacement from the set $\{z_1, \cdots, z_n\}$. Since the sampling was done with replacement, we have

$$E(y_i^2) = \sigma^2, \quad E(y_i y_j) = \frac{\sigma^2}{n},$$

$$E(\bar{y}^2) = \left(\frac{1}{mk} + \frac{1}{n} - \frac{1}{mnk}\right)\sigma^2,$$

$$E(y_i \bar{z}) = \frac{\sigma^2}{n}, \quad E(\bar{z}^2) = \frac{\sigma^2}{n},$$

where $\bar{y} = (y_1 + \cdots + y_{mk})/mk$ and $\bar{z} = (z_1 + \cdots + z_n)/n$. Then, by noting $V_{m,n}(\mathcal{D}|\boldsymbol{\theta}) = m\bar{z} + m\mu$, we have

$$E(l_{m,n,k}(\boldsymbol{D_s}|\boldsymbol{\theta}) - V_{m,n}(\mathcal{D}|\boldsymbol{\theta}))^2 = E(m\bar{y} - m\bar{z})^2 = \frac{m}{k}\left(1 - \frac{1}{n}\right)\sigma^2$$

Proof of Theorem 3.4.1 To prove this theorem, we introduce the following lemma which is due to Chen (1985).

Lemma A.0.1 *Let $\{f_n(\boldsymbol{x}), n = 1, 2, \cdots\}$ be a sequence of probability density functions defined on $\mathcal{X}$. Define $l_n = \log f_n(\cdot)$. It is assumed that, for each n, there exists a strict local maximum, $\boldsymbol{\mu}_n$, of f_n in $\mathcal{X}$ such that the following conditions are satisfied:*

(E_1) $l'_n(\boldsymbol{\mu}_n) = \partial l_n / \partial \boldsymbol{x}|_{\boldsymbol{x}=\boldsymbol{\mu}_n} = 0$.

(E_2) $l''_n(\boldsymbol{\mu}_n) = \partial^2 l_n / \partial \boldsymbol{x}\boldsymbol{x}^T\big|_{\boldsymbol{x}=\boldsymbol{\mu}_n}$ is negative definite; or $\Sigma_n = [-l''_n(\boldsymbol{\mu}_n)]^{-1}$ is positive definite.

(E_3) (Steepness) $\sigma_n \to 0$ as $n \to \infty$, where σ_n^2 is the largest eigenvalue of Σ_n defined in (E_2).

(E_4) (Smoothness) For any $\epsilon > 0$, there exists an integer N and $\eta > 0$ such that, for any $n > N$ and $\boldsymbol{x} \in H(\boldsymbol{\mu}_n, \eta) = \{\boldsymbol{x} \in \mathcal{X} : |\boldsymbol{x} - \boldsymbol{\mu}_n| < \eta\}$, $l_n''(\boldsymbol{x})$ exists and satisfies

$$I_d - A(\epsilon) \leq l_n''(\boldsymbol{x}) \left\{l_n''(\boldsymbol{\mu}_n)\right\}^{-1} \leq I_d + A(\epsilon),$$

where d is the dimension of $\boldsymbol{x}$, I_d is a $d \times d$ identity matrix, and $A(\epsilon)$ is a $d \times d$ positive semi-definite symmetric matrix whose largest eigenvalue tends to zero as $\epsilon \to 0$.

(E_5) (Concentration) For any $\eta > 0$, the probability

$$Q_n = \int_{H(\boldsymbol{\mu}_n, \eta)} f_n(\boldsymbol{x}) d\boldsymbol{x} \to 1, \quad as \quad n \to \infty.$$

Let X_n denote a sample of $f_n(\boldsymbol{x})$ and let $Z_n = \Sigma_n^{-1/2}(X_n - \boldsymbol{\mu}_n)$ Then Z_n converges in distribution to the standard normal Z whose pdf is $f(z) = (2\pi)^{-d/2} \exp\left\{-z^T z/2\right\}$.

Then, to prove Theorem 3.4.1, it suffices to verify that $\tilde{\pi}_m(\boldsymbol{\theta}|\mathcal{D})$ satisfies the conditions (E_1)-(E_5). The condition (D_2) implies that (E_1) and (E_2) are satisfied.

For $\tilde{\pi}_m(\boldsymbol{\theta}|\mathcal{D})$, the matrix corresponding to Σ_n in (E_2) is given by

$$\left[-\tilde{l}_m''(\boldsymbol{\mu}_m)\right]^{-1} = \frac{1}{m}\tilde{\Sigma}_{m,1},$$

where $\tilde{\Sigma}_{m,1}$ is positive definite and its eigenvalues are asymptotically independent of m. Hence, condition (E_3) is satisfied.

In condition (D_2), it is assumed that $\partial^2 \tilde{l}_m(\boldsymbol{\theta})/\partial\boldsymbol{\theta}\boldsymbol{\theta}^T$ is continuous on Θ. Therefore, the smoothness condition (E_4) is satisfied.

In condition (D_1), it is assumed that $\tilde{l}_m(\boldsymbol{\theta})$ is uniformly continuous on Θ, and it has a unique global maximum and a finite number of local maxima. Therefore, it follows from the existing result of simulated annealing, see e.g. Theorem 2.3 of Dekkers and Aarts (1991), that the concentration condition (E_5) is satisfied. This completes the proof of the theorem.